May, 1969

OPTIMALITY PROPERTIES OF VARIOUS PROCEDURES FOR RANKING n DIFFERENT NUMBERS USING ONLY BINARY COMPARISONS

by

Abdollah Hadian

Technical Report No. 117

University of Minnesota
Minneapolis, Minnesota

Submitted as a Thesis to the Faculty of the Graduate School of the University of Minnesota in partial fulfillment of the requirement for degree of Doctor of Philosophy.

Research supported by National Science Foundation Grant GP-9018.

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to Professor Milton Sobel for his expert guidance and encouragement during the preparation of this dissertation, and whose desire for research is inspiring. I would also like to express my thanks to Professor Robert J. Buehler for reading the manuscript and making helpful suggestions.

TABLE OF CONTENTS

TABLE OF CONTENTS

Pages

ABSTRACT

OPTIMALITY PROPERTIES OF VARIOUS PROCEDURES

FOR RANKING n DIFFERENT NUMBERS

USING ONLY BINARY COMPARISONS

There are n (≥ 2) unknown real numbers which are pairwise unequal. Starting from a random order, we want to rank the t largest ($1 \leq t \leq n-1$) of these numbers by using only binary errorless comparisons. After each comparison, the experimenter is only told which is larger (or equivalently, smaller). The problem is to obtain a procedure for finding and ordering the t largest numbers which is such that the (random) number of comparisons is small in some well-defined sense.

The main part of this work is concerned with the case $t = n - 1$. Two optimality criteria are introduced: to minimize the expected number of comparisons and to minimize the maximum number of comparisons. It is found that

$$H(n!) = \{\log n!\} - \left(\frac{2^{\{\log n!\}} - n!}{n!}\right)$$

and $\{H(n!)\}$ are lower bounds for the first and second criterion, respectively, under any procedure. (Here all logs are to base 2 and $\{x\}$ is the smallest integer greater than or equal to x.) All the procedures whose expectation achieve the lower bound $H(n!)$ are called E-noiseless and are characterized in theorem 1a by the fact that the range (or difference between the maximum and minimum number of comparisons) is exactly one. It is proved in theorem 2a that the maximum number of

Abdellah HAMIAN-Bohkordt
250 words

ABSTRACT

OPTIMALITY PROPERTIES OF VARIOUS PROCEDURES FOR RANKING n DIFFERENT NUMBERS USING ONLY BINARY COMPARISONS

There are n (≥ 2) unknown real numbers which are all unequal. Starting from a random order, we want to rank the t largest ($1 \leq t \leq n-1$) of these numbers by using only binary costless comparisons. After each comparison, the experimenter is only told which is larger (or equivalently smaller). The problem is to obtain a procedure for finding and ordering the t largest numbers which is such that the (random) number of comparisons is small in some well-defined sense.

The main part of this work is concerned with the case $t = n - 1$. Two optimality criteria are introduced: to minimize the expected number of comparisons and to minimize the maximum number of comparisons. It is found that

$$H(n!) = \{\log n!\} - \frac{2^{\{\log n!\}} - n!}{n!}$$

and $\{L(n!)\}$ are lower bounds for the first and second criterion, respectively. (Here all logs are to base 2 and $\{x\}$ is the smallest integer greater than or equal to x.) All the procedures whose expectation achieve the lower bound $H(n!)$ are called H-procedures and are characterized in Theorem 1 by the fact that the range (or difference between the maximum and minimum number of comparisons) is exactly one. It is proved in Theorem 2 that the maximum number of

comparisons of E-noiseless procedures achieve the lower bound $\{H(n!)\}$, but the converse need not be true.

In Chapter 2 a certain subclass $\mathcal{I}$ of inductive procedures is introduced and all the optimal procedures in $\mathcal{I}$ are found (Theorems 3 and 4). In particular, it is noted that the Steinhaus procedure R_S is one of the optimal procedures in the subclass $\mathcal{I}$.

In Chapter 3 a procedure R_{FJ} due to Ford and Johnson (see reference 5) is studied. An explicit expression for the maximum number of comparisons under R_{FJ} is found and it is shown that this maximum is strictly smaller than the maximum under R_S for $n \geq 5$. A subclass of procedures $\mathcal{J}$, which includes R_{FJ}, is considered and lower bounds for all procedures in $\mathcal{J}$ are found. For $n = 7$ an optimal procedure in the subclass $\mathcal{J}$ is given which proves that R_{FJ} does not have minimum expectation in the subclass $\mathcal{J}$, let alone among all procedures.

Milton Sobel

CHAPTER I

General Results for the Problem of Ranking n Items

1.1. Statement of the problem.

There are n (an integer ≥ 2) unknown real numbers $x_1, x_2, \ldots x_n$, each called an item, which are pairwise unequal. Starting from a random order we want to rank the t largest $(1 \leq t \leq n-1)$ of these items by means of a sequence of binary comparisons. After each comparison the experimentor is told merely which of the two items compared is larger (or equivalently smaller). The problem is to find a procedure R for ranking (i.e., finding and ordering) the t largest items in such a way that the required number of comparisons should be small in some well-defined sense.

The items $x_1, x_2, \ldots, x_n$ can be regarded as different comparable quantities such as weights or abilities of n players in a tournament in which each game involves two opposing players (such as tennis). In the first instance comparisons are made by means of a two-pan scale, allowing only one object in each pan; in the second instance the comparisons are made by having two players play against each other. We assume that the better player always wins; so that no game can end in a draw and transitivity is used whenever possible. We also assume that our procedures will not include any comparison for which the result is known or can be inferred from known results.

Special cases of this problem are:

(a) $t = 1$, which is relatively trivial and in the literature is mostly referred to as a knock-out tournament (see, for example, [4]).

(b) $t = 2$, which is studied by M. Sobel [16].

CHAPTER I

General Results for the Problem of Ranking n Items

1.1. Statement of the problem.

There are n (an integer ≥ 2) real numbers $x_1, x_2, \ldots, x_n, \ldots$ each called an item, which are pairwise unequal. Starting from a random order we want to rank the t largest $(1 \leq t \leq n-1)$ of these items by means of a sequence of binary comparisons. After each comparison the experimenter is told merely which of the two items compared is larger (or equivalently smaller). The problem is to find a procedure R for ranking (i.e., finding and ordering) the t largest items in such a way that the required number of comparisons should be small in some well-defined sense.

The items $x_1, x_2, \ldots, x_n$ can be regarded as different comparable quantities such as weights or abilities of n players in a tournament in which each game involves two opposing players (such as tennis). In the first instance comparisons are made by means of a two-pan scale, allowing only one object in each tray; in the second instance the comparisons are made by having two players play against each other. We assume that the better player always wins; so that no game can end in a draw and transitivity is used whenever possible. We also assume that our procedures will not include any comparison for which the result is known or can be inferred from known results.

Special cases of this problem are:

(a). $t = 1$, which is relatively trivial and in the literature is mostly referred to as a knock-out tournament (see, for example, [illegible]).

(b). $t = 2$, which is studied by M. Sobel [10].

(c) $t = n - 1$, i.e., ranking all items using only binary comparisons.

The main part of this work is concerned with the case $t = n - 1$; nevertheless the concepts and notations are introduced for the case of general t.

The number of comparisons, under any procedure, required to rank the t largest of n items is a discrete random variable (possibly degenerate in some cases). For example, if $n = 3$ and $t = 2$ any procedure requires 2 or 3 comparisons with probability 1/3 or 2/3, respectively. Thus we may consider two optimality properties described in the following sections.

1.2 E-optimality.

Let $A_t(n|R)$ be the expected number of comparisons required to rank the t largest of n items under a procedure R. Let

$$(1.1) \qquad L(t, n) = \min_{R \in \mathcal{R}_t} A_t(n|R)$$

where R is a procedure in the (finite) class $\mathcal{R}_t$ of all possible procedures. For $t = n - 1$ we denote $A_{n-1}(n|R)$ by $A(n|R)$, $\mathcal{R}_{n-1}$ by $\mathcal{R}$ and $L(n-1, n)$ by $L(n)$.

A procedure R is called E-optimal if for each n and t

$$(1.2) \qquad A_t(n|R) = L(t, n).$$

Since the result of any comparison of two items is the knowledge of one being greater or smaller than the other, a procedure can be represented as a tree (or a directed graph) without any loops and having exactly two forks at each node, we refer to this simply as a tree.

(c) $t = n - 1$, i.e., ranking all items using only binary comparisons.

The main part of this work is concerned with the case $t = n - 1$; nevertheless the concepts and notations are introduced for the case of general t.

The number of comparisons, under any procedure, required to rank the t largest of n items is a discrete random variable (possibly degenerate in some cases). For example, if $n = 3$ and $t = 2$ any procedure requires 2 or 3 comparisons with probability 1/3 or 2/3, respectively. Thus we may consider two optimality properties described in the following sections.

1.2 A-optimality.

Let $A_t(n|R)$ be the expected number of comparisons required to rank the t largest of n items under a procedure R. Let

$$L(t, n) = \operatorname*{Min}_{R \in \mathcal{P}_t} A_t(n|R) \tag{1.1}$$

where R is a procedure in the (finite) class $\mathcal{P}_t$ of all possible procedures. For $t = n - 1$ we denote $A_{n-1}(n|R)$ by $A(n|R)$, $\mathcal{P}_{n-1}$ by $\mathcal{P}$ and $L(n-1, n)$ by $L(n)$.

A procedure R is called A-optimal if for each n and t

$$A_t(n|R) = L(t, n). \tag{1.2}$$

Since the result of any comparison of two items is the knowledge of one being greater or smaller than the other, a procedure can be represented as a tree (or a directed graph) without any loops and having exactly two forks at each node, we refer to this simply as a tree.

$A_t(n|R)$ then corresponds to the average (branch) length of the tree that represents a procedure. For example a procedure for ranking 3 items ($n = 3$, $t = 2$) is given by the tree in Fig. 1. For any of our trees, the arrow pointed to the left under x_i vs. x_j (to be read x_i versus x_j) indicates the result that $x_i < x_j$. Similarly the arrow pointed to the right under x_i vs. x_j indicates the result that $x_i > x_j$.

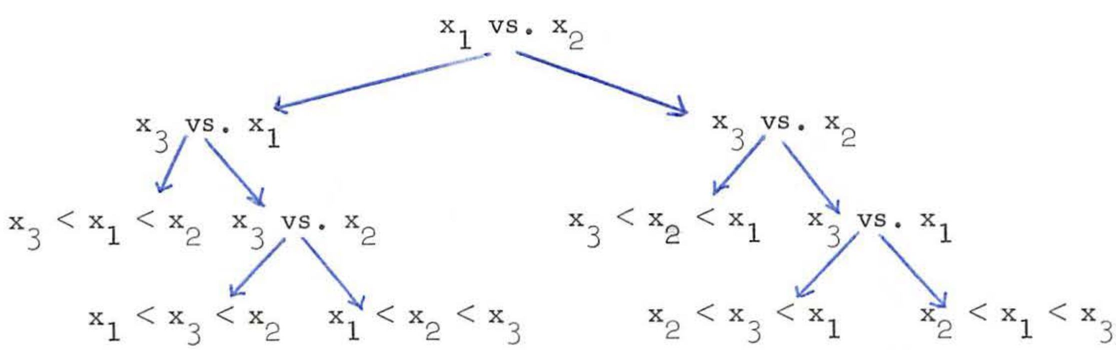


Fig. 1. A procedure for ranking 3 items; x_1, x_2 and x_3.

The end points of this tree correspond to the $3! = 6$ permutations of x_1, x_2 and x_3. For the procedure R_0 given in Fig. 1 we have $A(3|R_0) = (2)\frac{2}{6} + (3)\frac{4}{6} = \frac{8}{3}$.

Procedures that have the same expected number of comparisons are called E-equivalent. The E-excess of a procedure R, denoted by $C_t(n|R)$, is defined to be

$$C_t(n|R) = A_t(n|R) - L(t, n) \tag{1.3}$$

where $L(t, n)$ is defined by (1.1). $L(t, n)$ in general is unknown. Special cases for which $L(t, n)$ is known or conjectured to be known are the following:

$A_t(n|K)$ then corresponds to the average (branch) length of the tree that represents a procedure. For example a procedure for ranking 3 items ($n = 3$, $t = 2$) is given by the tree in Fig. 1. For any of our trees, the arrow pointed to the left under x_i vs. x_j (to be read x_i versus x_j) indicates the result that $x_i < x_j$. Similarly the arrow pointed to the right under x_i vs. x_j indicates the result that $x_i > x_j$.

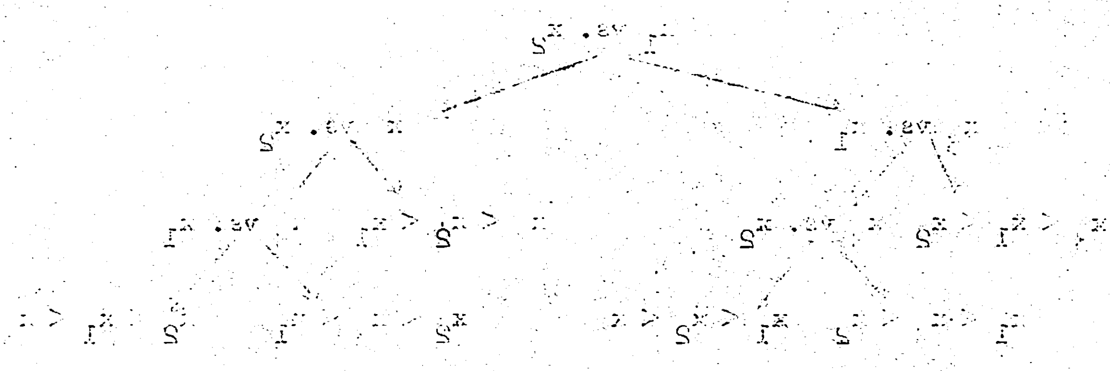


Fig. 1. A procedure for ranking 3 items: x_1, x_2 and x_3.

The end points of this tree correspond to the $3! = 6$ permutations of x_1, x_2 and x_3. For the procedure K_0 given in Fig. 1 we have

$$A(3|K_0) = (2)\frac{2}{6} + (3)\frac{4}{6} = [illegible].$$

Procedures that have the same expected number of comparisons are called *K-equivalent*. The *K-excess* of a procedure K, denoted by $C_t(n|K)$, is defined to be

$$(1.3) \qquad C_t(n|K) = A_t(n|K) - L(t, n)$$

where $L(t, n)$ is defined by (1.1). $L(t, n)$ in general is unknown. Special cases for which $L(t, n)$ is known or conjectured to be known are the following:

(a) $t = 1$, where $L(1, n) = n - 1$ (see, for example, [4] and [17]).

(b) $t = 2$, where $L(2, n)$ is conjectured to be $n + r - 2$ for $n = 2^r$ (see [16]).

(c) $t = n - 1$, in this case

$$L(n) = H(n!) \quad \text{for} \quad n \leq 6,\ n = 9,\ 10 \tag{1.4}$$

where for any integer $m \geq 1$

$$H(m) = [\log m] + 2\left(\frac{m-2^{[\log m]}}{m}\right) = \{\log m\} - \frac{2^{\{\log m\}}-m}{m}. \tag{1.5}$$

Here $[x]$ denotes the largest integer not greater than x, and $\{x\}$ denotes the smallest integer not less than x. (All logarithms used in this work are to the base 2 unless otherwise stated.) Picard [11] has found the values of $L(n)$ for $n \leq 6$, and Y. Cesari (private communication) has found $L(n)$ for $n = 9, 10$, both by exhibiting trees whose expected length is equal to $H(n!)$ for $n \leq 6$ and $n = 9, 10$. $H(m)$ given in (1.5) is called the Huffman lower bound (see [11], [14], [15] and [16]). Other equivalent forms of $H(m)$ which will be used in our later discussion are:

$$\begin{aligned} H(m) &= r + \frac{2c}{m} \\ H(m) &= s - \frac{d}{m} \end{aligned} \tag{1.6}$$

where non-negative integers $r = r_m$, $c = c_m$, $s = s_m$ and $d = d_m$ are defined by

$$\begin{aligned} m &= 2^r + c \qquad 0 \leq c < 2^r \\ m &= 2^s - d \qquad 0 \leq d < 2^{r-1}. \end{aligned} \tag{1.7}$$

(a) $t = 1$, where $L(1, n) = n - 1$ (see, for example, [8] and [17]).

(b) $t = 2$, where $L(2, n)$ is conjectured to be $n + t - 2$ for $n = 2^t$ (see [16]).

(c) $t = n - 1$, in this case

$$L(n) = H(n!) \quad \text{for } n \leq 6,\ n = 9, 10. \tag{1.4}$$

where for any integer $m \geq 1$

$$H(m) = \lfloor \log m \rfloor + 2\left(\frac{m - 2^{\lfloor \log m \rfloor}}{m}\right) = \lceil \log m \rceil - \frac{2^{\lceil \log m \rceil} - m}{m}. \tag{1.5}$$

Here $\lfloor x \rfloor$ denotes the largest integer not greater than x, and $\lceil x \rceil$ denotes the smallest integer not less than x. (All logarithms used in this work are to the base 2 unless otherwise stated.) Ficard [11] has found the values of $L(n)$ for $n \leq 6$, and M. Cesari (private communications) has found $L(n)$ for $n = 9, 10$. Both by exhibiting trees whose expected length is equal to $H(n!)$ for $n \leq 6$ and $n = 9, 10$. $H(m)$ given in (1.5) is called the Huffman lower bound (see [11], [14], [17] and [16]). Other equivalent forms of $H(m)$ which will be used in our later discussion are:

$$(1.6)\qquad \begin{aligned} H(m) &= r + \frac{2c}{m} \\ H(m) &= s - \frac{q}{m} \end{aligned}$$

where non-negative integers $r = r_m$, $c = c_m$, $s = s_m$ and $q = q_m$ are defined by

$$(1.7)\qquad \begin{aligned} m &= 2^r + c \qquad 0 \leq c < 2^r \\ m &= 2^s - q \qquad 0 \leq q < 2^{s-1}. \end{aligned}$$

The E-optimality of any procedure with expected number of comparisons equal to $H(n!)$ is proved in lemma 1 below. We mention that the procedure given by Steinhaus [18] which is identical with the R_{FJ} procedure [5] for $n = 5$ is E-optimal even though it was introduced to have a different optimality (see next section) property.

The E-noise $B(n|R)$ of a procedure R is defined by

$$(1.8) \qquad B(n|R) = A(n|R) - H(n!).$$

The procedure R is called E-noiseless if $B(n|R) = 0$. It should be noted that an E-noiseless procedure may not exist whereas an E-optimal procedure always exists.

Lemma 1. If a procedure is E-noiseless then it is E-optimal.

Proof. First we show that

$$(1.9) \qquad H(n!) \leq L(n) \leq A(n|R).$$

In fact, for any problem with m states of nature, one of which is true, $H(m)$ represents the minimum expected number of (yes or no) questions need to find the true one when there is no restriction whatever on the question that can be asked, i.e., we can choose any subset and ask if the true state is contained in it (see [11], [12] and [14]). On the other hand in our problem of ranking n items $(t = n - 1)$ we have $m = n!$ states of nature and we can only ask questions (or make partitions) that correspond to some binary comparison. This already proves $H(n!) \leq A(n|R)$ and $H(n!) \leq L(n)$. Hence (1.9) follows easily.

Now consider any procedure R, for our problem, which is E-noiseless. Then

The N-optimality of any procedure with expected number of comparisons equal to $H(n!)$ is proved in the lemma below. We mention that the procedure given by Steinhaus [13] which is identical with the T_n procedure [ ] for $n =$ is N-optimal even though it was introduced to have a different optimality (see next section) property.

The N-noise $E(n|K)$ of a procedure K is defined by

$$E(n|K) = A(n|K) - H(n!). \tag{1.8}$$

The procedure K is called N-noiseless if $E(n|K) = 0$. It should be noted that an N-noiseless procedure may not exist whereas an N-optimal procedure always exists.

Lemma 1. If a procedure is N-noiseless then it is N-optimal.

Proof. First we show that

$$H(n!) \leq B(n) \leq A(n|K). \tag{1.9}$$

In fact, for any problem with m states of nature, one of which is true, $H(\cdot)$ represents the minimum expected number of (pressure) questions need to find the true one when there is no restriction whatever on the question that can be asked, i.e., we can choose any subset and ask if the true state is contained in it (see [11], [12] and [14]). On the other hand in our problem of ranking n items ($t = n - 1$) we have $m = n!$ states of nature and we can only ask questions (or make partitions) that correspond to some binary comparison. This already proves $H(n!) \leq A(n|K)$ and $H(n!) \leq B(n)$, hence (1.9) follows easily.

Now consider any procedure K for our problem, which is N-noiseless. Then

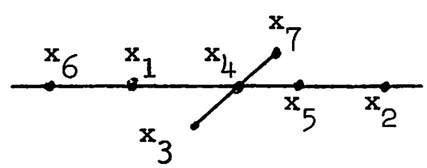
$$(1.10) \qquad 0 = A(n|R) - H(n!) \geq A(n|R) - L(n) \geq 0$$

and hence $A(n|R) = L(n)$, i.e., R is E-optimal.

To briefly illustrate the fact that certain partitions do not correspond to a comparison we assume that we wish to rank 7 items $x_1, x_2, \ldots, x_7$ which are known to satisfy

$$(1.11) \qquad x_6 < x_1 < x_4 < x_5 < x_2 \quad \text{and} \quad x_3 < x_4 < x_7.$$

These inequalities in (1.11) can also be shown by the following diagram



Here there are $3 \cdot 3 = 9$ states of nature which are all possible permutations of 7 items consistent with (1.11). The next comparison has to be x_3 vs. x_6 or x_3 vs. x_1 or x_7 vs. x_5 or x_7 vs. x_2 and these partition the 9 possible states of nature into subgroups of sizes (3, 6), (6, 3), (3, 6) and (6, 3), respectively. In particular we note that there is no comparison which corresponds to the 'crucial' partition (4, 5), (nor to any of the other partitions such as (8, 1), (7, 2) etc.). Suppose we want to rank 7 items satisfying the inequalities (1.11). It can be easily checked that in this case any procedure is as good as the other and requires, on the average, $2((1)\frac{1}{3} + (2)\frac{2}{3}) = 3 + \frac{1}{3}$ comparisons. Whereas if the partition (4, 5) would have been allowed we would have needed, in average, $1 + \frac{4}{9} H(4) + \frac{5}{9} H(5) = 3 + \frac{2}{9} = H(9)$ questions.

The above lemma and illustration indicate that a procedure may simultaneously have positive E-noise and be E-optimal. In fact we

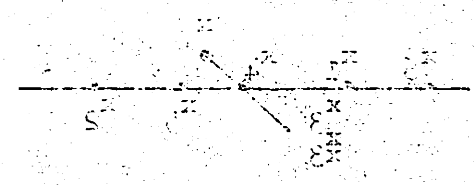
(1.10) $0 = A(n|R) - E(n_i) \geq A(n|R) - L(n) \geq 0$

and hence $A(n|R) = L(n)$, i.e., R is L-optimal.

To briefly illustrate the fact that certain partitions do not correspond to a comparison we assume that we wish to rank 7 items $x_1, x_2, \ldots, x_7$ which are known to satisfy

(1.11) $x_6 < x_1 < x_4 < x_3 < x_2$ and $x_5 < x_4 < x_7$.

These inequalities in (1.11) can also be shown by the following diagram



Here there are [illegible] $= 9$ states of nature which are all possible permutations of 7 items consistent with (1.12). The next comparison has to be $x_{[illegible]}$ vs. x_6 or $x_{[illegible]}$ vs. x_1 or $x_{[illegible]}$ vs. x_4 or $x_{[illegible]}$ vs. $x_{[illegible]}$ and these partition the 9 possible states of nature into subgroups of sizes ([illegible]), (6, [illegible]), ([illegible]) and ([illegible]), respectively. In particular we note that there is no comparison which corresponds to the 'crucial' partition (4, 5), (nor to any of the other partitions such as (8, 1), (7, 2) etc.). Suppose we want to rank 7 items satisfying the inequalities (1.11). It can be easily checked that in this case any procedure is as good as the other and requires, in average, $2((1)\frac{[illegible]}{9} + (2)\frac{[illegible]}{9}) = 3 + \frac{[illegible]}{9}$ comparisons. Whereas if the partition (4, 5) would have been allowed we would have needed, in average,

$1 + \frac{4}{9}E(4) + \frac{5}{9}E(5) = 3 + \frac{5}{9} = E(9)$ questions.

The above lemma and illustration indicate that a procedure may simultaneously have positive I-noise and be L-optimal. In fact we conjecture that conjecture that the procedure R_1 given below for $n = 7$ (starting from a random order) is such a procedure. The E-noise for R_1 given by

$$(1.12) \qquad B(7|R_1) = 12 + \frac{121}{315} - H(7!) = \frac{1}{105} = .0095.$$

Cesari [2] has proved that it is impossible to have a procedure for ranking 7 items which has zero E-noise. He has given a procedure equivalent to R_1 (i.e., with the same amount of noise but not identical with it).

For procedure R_1 we start with the following comparisons: x_1 vs. x_2; x_3 vs. x_4; x_5 vs. x_6, assuming, without loss of generality, that $x_1 < x_2$, $x_3 < x_4$ and $x_5 < x_6$. Then we compare x_2 vs. x_4 and assume $x_2 < x_4$. The continuation of the procedure R_1, after these 4 comparisons is represented by the tree in Fig. 2 (further explanations of the numbers shown are given in the appendix).

1.3. M-optimality (minimax).

Let $M_t(n|R)$ be the maximum number of comparisons required to find and order the t largest of n items under a procedure R. Let

$$(1.13) \qquad L_{\text{Max}}(t, n) = \min_{R \in \mathcal{R}_t} M_t(n|R)$$

where R is a procedure belonging to the (finite) class $\mathcal{R}_t$ of all possible procedures. For $t = n - 1$ we denote $M_{n-1}(n|R)$ by $M(n|R)$, $\mathcal{R}_{n-1}$ by $\mathcal{R}$ and $L_{\text{Max}}(n-1, n)$ by $L_{\text{Max}}(n)$.

A procedure R is called M-optimal if for each n and t

$$(1.14) \qquad M_t(n|R) = L_{\text{Max}}(t, n).$$

conjectures that the procedure R_1 given below for $n = 7$ (starting from a random order) is such a procedure. The E-noise for R_1 given by

$$(1.12) \qquad E(7|R_1) = 12 + \frac{121}{[illegible]} - H(7) = \frac{1}{10[illegible]} = .00[illegible]..$$

Césari [2] has proved that it is impossible to have a procedure for ranking 7 items which has zero E-noise. He has given a procedure equivalent to R_1 (i.e., with the same amount of noise but not identical with it).

For procedure R_1 we start with the following comparisons: x_1 vs. x_2; x_3 vs. x_4; x_5 vs. x_6, assuming without loss of generality that $x_1 < x_2$, $x_3 < x_4$ and $x_5 < x_6$. Then we compare x_2 vs. x_4 and assume $x_2 < x_4$. The continuation of the procedure R_1 after these 4 comparisons is represented by the tree in Fig. 2 (further explanations of the numbers shown are given in the appendix).

1.4. M-optimality (minimax).

Let $M_t(n|R)$ be the maximum number of comparisons required to find and rank the t largest of n items under a procedure R. Let

$$(1.13) \qquad L_{max}(t, n) = \underset{R \in \mathcal{P}_t}{\mathrm{Min}}\, M_t(n|R)$$

where R is a procedure belonging to the (finite) class $\mathcal{P}_t$ of all possible procedures. For $t = n - 1$ we denote $M_{n-1}(n|R)$ by $M(n|R)$, $\mathcal{P}_{n-1}$ by $\mathcal{P}$ and $L_{max}(n-1, n)$ by $L_{max}(n)$.

A procedure R is called M-optimal if for each t and n

$$(1.14) \qquad M_t(n|R) = L_{max}(t, n).$$

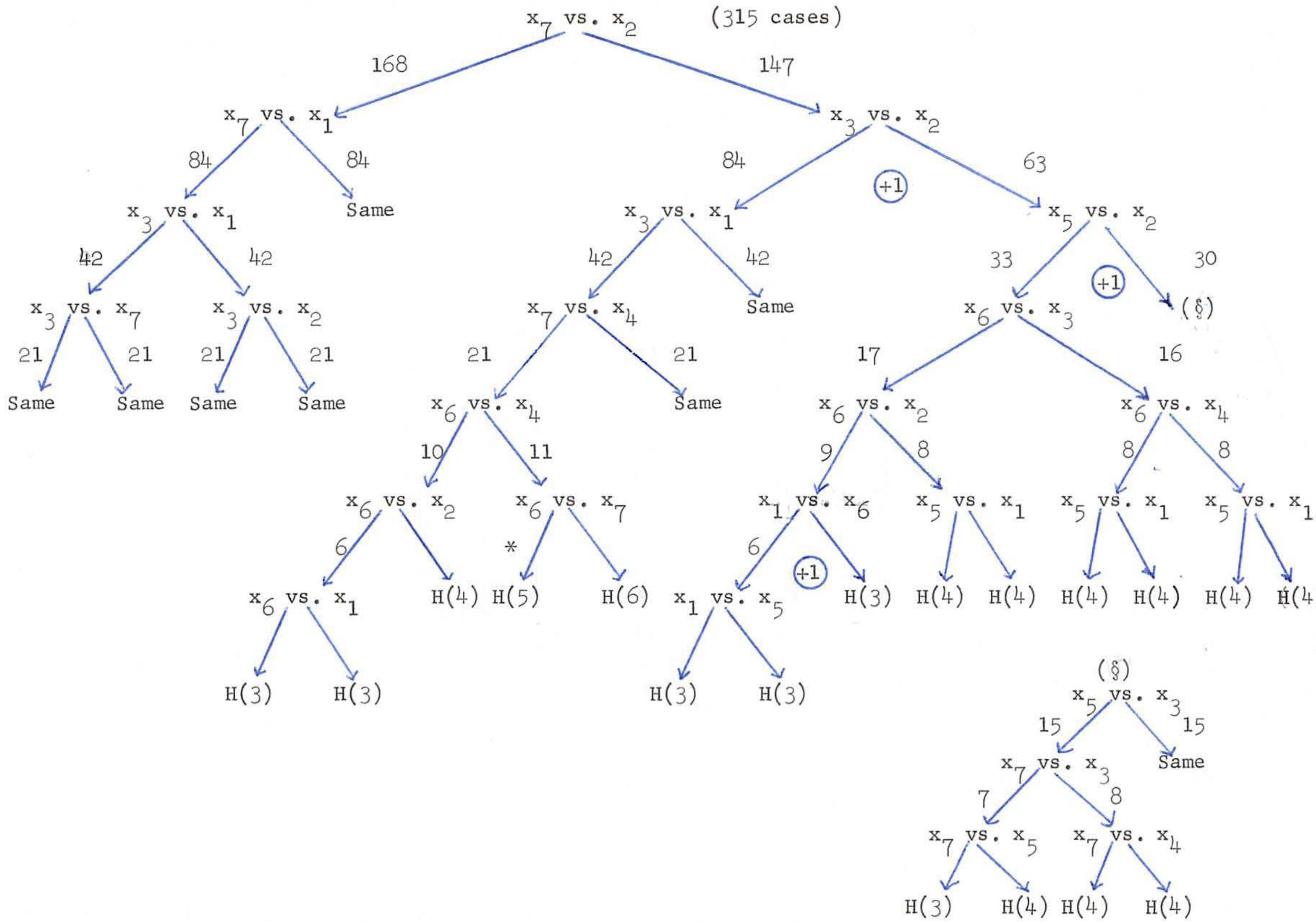


Fig. 2. A tree for the continuation of procedure R_1 for ranking 7 items conjectured to be E-optimal.

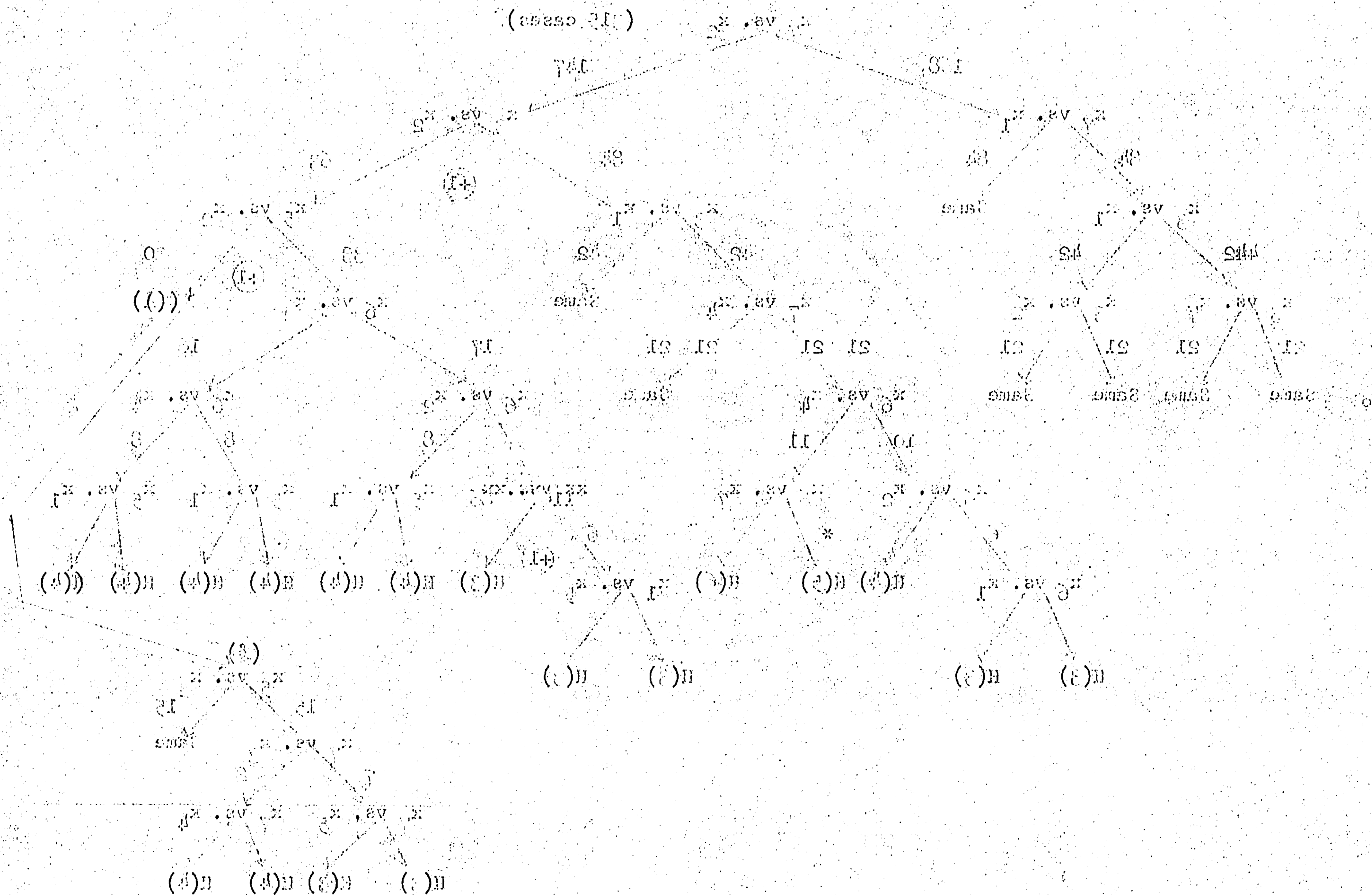


Fig. 2. A tree for the construction of procedure R_1 for ranking 7 items conjectured to be [illegible]-optimal.

All procedures having the same maximum number of comparisons are called M-equivalent. When the complete procedure R is written as a tree, $M_t(n|R)$ corresponds to the maximum (branch) length of this tree.

The M-excess of a procedure R, denoted by $D_t(n|R)$, is defined to be

$$(1.14a)\qquad D_t(n|R) = M_t(n|R) - L_{Max}(t, n).$$

In general $L_{Max}(t, n)$ is unknown; special cases in which $L_{Max}(t, n)$ is known are the following:

(a) $t = 1$, where $L_{Max}(1, n) = L(1, n) = n - 1$.

(b) $t = 2$, where $L_{Max}(2, n) = n - 2 + \{\log n\}$ (see [13], [16] and reference 20 in [16]). We notice from (b) in Section 1.2 that

$$(1.15)\qquad L_{Max}(2, n) = L(2, n) = n + r - 2 \quad \text{for} \quad n = 2^r.$$

M. Sobel has introduced several procedures for this case, two of which achieve $L_{Max}(2, n)$ and hence are M-optimal.

(c) $t = n - 1$ for $n \leq 11$ and $n = 20, 21$. It is shown by Ford and Johnson [5] that

$$(1.16)\qquad L_{Max}(n) = \{\log n!\} \quad \text{for} \quad n \leq 11, \text{ and } \quad n = 20, 21.$$

They have found $L_{Max}(n)$ by introducing a specific procedure R_{FJ}, for all n, which attains the lower bound in (1.16), i.e.,

$$(1.17)\qquad M(n|R_{FJ}) = \{\log n!\} \quad \text{for} \quad n \leq 11, \text{ and } \quad n = 20, 21.$$

The fact that $\{\log n!\}$ is a lower bound for the maximum number of comparisons under any procedure is proved in [5] and [9a-Theorem 24], and

All procedures having the same maximum number of comparisons are called M-equivalent. When the complete procedure P is written as a tree, $M_P(n|K)$ corresponds to the maximum (branch) length of this tree.

The M-excess of a procedure P, denoted by $D_P(n|K)$, is defined to be

$$(1.14a) \qquad D_P(n|K) = M_P(n|K) - L_{Max}(t, n).$$

In general $L_{Max}(t, n)$ is unknown; special cases in which $L_{Max}(t, n)$ is known are the following:

(a) $t = 1$, where $L_{Max}(1, n) = L(1, n) = n - 1$.

(b) $t = 2$, where $L_{Max}(2, n) = n - 2 + \lceil \log_2 n \rceil$ (see [1.1]) [16] and reference 20 in [15]). We notice from (b) in Section 1.2 that

$$(1.15) \qquad L_{Max}(2, n) = L(2, n) = n + t - 2 \quad \text{for } n = 2^t.$$

M. Sobel has introduced several procedures for this case, two of which achieve $L_{Max}(2, n)$ and hence are M-optimal.

(c) $t = n - 1$ for $n \leq 11$ and $n = 20, 21$. It is shown by Ford and Johnson [5] that

$$(1.16) \qquad L_{Max}(n) = \lceil \log_2 n! \rceil \quad \text{for } n \leq 11, \text{ and } n = 20, 21.$$

They have found $L_{Max}(n)$ by introducing a specific procedure P_{FJ} for all n, which attains the lower bound in (1.10), i.e.,

$$(1.17) \qquad M(n|K_{FJ}) = \lceil \log n! \rceil \quad \text{for } n \leq 11, \text{ and } n = 20, 21.$$

The fact that $\lceil \log n! \rceil$ is a lower bound for the maximum number of comparisons under any procedure is explained briefly in [5] and proved

proved at length in [9, Theorem 1]. We now give another proof in the following lemma based on (1.9).

Lemma 2. For any procedure R we have

$$\{\log n!\} = \{H(n!)\} \leq L_{Max}(n) \leq M(n|R). \tag{1.18}$$

Proof. From (1.9) we have for any procedure R

$$H(n!) \leq A(n|R) \leq M(n|R). \tag{1.19}$$

In particular letting R be the M-optimal procedure we have

$$H(n!) \leq L_{Max}(n). \tag{1.20}$$

Since $M(n|R)$ and $L_{Max}(n)$ are integers and since $L_{Max}(n) \leq M(n|R)$, using (1.19) and (1.20) we have

$$\{H(n!)\} \leq L_{Max}(n) \leq M(n|R). \tag{1.21}$$

To finish the proof of the lemma we note that for any integer m with $2^{s-1} < m \leq 2^s$ we have $H(m) = s - \theta$ where $0 \leq \theta < 1$. It follows that $\{H(m)\} = s = \{\log m\}$.

The M-noise $B_{Max}(n|R)$ of a procedure R is defined by

$$B_{Max}(n|R) = M(n|R) - \{H(n!)\}. \tag{1.22}$$

The procedure R is said to be M-noiseless if its M-noise is zero. It follows from (1.21) that an M-noiseless procedure is M-optimal, but the converse does not necessarily hold. In fact the case $n = 12$ is an example of this since we note that $\{H(12!)\} = 29$ and it has been shown by M. B. Wells (private communication) that $L_{Max}(12) \geq 30$. Since at length in [9, Theorem 1]. We now give another proof in the following Lemma based on (1.19), or (1.9).

Lemma 2. For any procedure R we have

$$(1.18)\qquad \lceil \log n! \rceil = \lceil H(n!) \rceil \le L_{Max}(n) \le M(n|R).$$

Proof. From (1.9) we have for any procedure R

$$(1.19)\qquad H(n!) \le A(n|R) \le M(n|R).$$

In particular letting R be the M-optimal procedure we have

$$(1.20)\qquad H(n!) \le L_{Max}(n).$$

Since $M(n|R)$ and $L_{Max}(n)$ are integers and since $L_{Max}(n) \le M(n|R)$, using (1.19) and (1.20) we have

$$(1.21)\qquad \lceil H(n!) \rceil \le L_{Max}(n) \le M(n|R).$$

To finish the proof of the lemma we note that for any integer n with $2^{s-1} < n \le 2^s$ we have $H(n) = s - \theta$ where $0 \le \theta < 1$. It follows that $\lceil H(n) \rceil = s = \lceil \log n \rceil$.

The M-noise $ZB'_{Max}(n|R)$ of a procedure R is defined by

$$(1.22)\qquad ZB'_{Max}(n|R) = M(n|R) - \lceil H(n!) \rceil.$$

The procedure R is said to be M-noiseless if its M-noise is zero. It follows from (1.21) that an M-noiseless procedure is M-optimal, but the converse does not necessarily hold. In fact the case $n = 12$ is an example of this since we note that $\lceil H(12!) \rceil = 29$ and it has been shown by Wells (private communication) that $L_{Max}(12) \ge 30$. Since

$M(n|R_{FJ}) = 30$, it follows that R_{FJ} is M-optimal even though it has one unit of M-noise.

To investigate the properties of M-noiseless procedures we need to solve an auxiliary problem which is of some interest per se.

Auxiliary Problem: There are $N \geq 2$ objects or possible states of nature. Each of these has probability $\frac{1}{N}$ of being the true state of nature which is unknown to us. We are to identify the true state of nature by a sequence of questions. We assume that after i questions $(i = 0, 1, 2, \ldots)$ the true state of nature is among N_i states, where $N = N_0 > N_1 > N_2 > \ldots > 1$. If $N_i = 1$ for any i the true state of nature is identified; otherwise we partition the N_i states arbitrarily into two disjoint sets and ask in which of these two sets the true states of nature lies. The answer is given correctly. The problem is to find a procedure $T_N^{(M)}$ that minimizes the maximum number of questions. It is also of interest to find $G(N)$, the maximum number of questions asked under the procedure $T_N^{(M)}$.

Remark 1. A closely related problem would arise if we are interested in a procedure which minimizes the expected number of questions. This is a solved problem and is treated in [11], [12], [14] and [16]; we used this result in the proof of Lemma 1 above. In fact, let $T_m^{(E)}$ be a procedure that minimizes the expected number of questions given that the true state is among m objects $(2 \leq m \leq N)$. Then, as in [16] under $T_m^{(E)}$ we select a set of size $y = y(m)$ for the next questions where y is any integer such that there is no power of 2 between y and $m - y$, i.e., if

$$(1.23) \qquad \Delta_m^{(E)} = \{j \mid \text{there is no power of 2 strictly between the integers } j \text{ and } m - j\}$$

$M(n|R_{11}) = K_0$. It follows that R_{11} is M-optimal even though it has one unit of M-noise.

To investigate the properties of M-noiseless procedures we need to solve an auxiliary problem which is of some interest per se.

Auxiliary Problem: There are $N \geq 2$ objects or possible states of nature. Each of these has probability $\frac{1}{N}$ of being the true state of nature which is unknown to us. We are to identify the true state of nature by a sequence of questions. We assume that after i questions $(i = 0, 1, 2, \dots)$ the true state of nature is among N_i states, where $N = N_0 > N_1 > N_2 > \dots \geq 1$. If $N_i = 1$ for any i the true state of nature is identified; otherwise we partition the N_i states arbitrarily into two disjoint sets and ask in which of these two sets the true state of nature lies. The answer is given correctly. The problem is to find a procedure $R_N^{(M)}$ that minimizes the maximum number of questions. It is also of interest to find $C(N)$, the maximum number of questions asked under the procedure $T_N^{(M)}$.

Remark 1. A closely related problem would arise if we are interested in a procedure which minimizes the expected number of questions. This is a solved problem and is treated in [11], [12], [13] and [16]; we used this result in the proof of Lemma 1 above. In fact, let $T_n^{(E)}$ be a procedure that minimizes the expected number of questions given that the true state is among n objects $(2 \leq n \leq N)$. Then, as in [13] under $T_n^{(E)}$ we select a set of size $y = y(n)$ for the next questions where y is any integer such that there is no power of 2 between y and $n - y$, ibid., 23

$$(E) = \{\text{there is no power of 2 strictly between the integers } y \text{ and } n - y\} \qquad (1.23)$$

we select any y in $\Delta_m^{(E)}$. It is easy to see that $\Delta_m^{(E)}$ is never empty and consists of a single integer if and only if m is a power of 2. The expected number of questions under $T_m^{(E)}$ is $H(m)$ given by (1.5).

Remark 2. The auxiliary problem and also the expectation problem in remark 1 are relatively easy problems because after each question we are left with a situation entirely similar to the original one. It should be noted that there are no restictions on the size or nature of the set that we can choose for the question. This is unlike the problem stated in Section 1.1, say for $t = n - 1$, where the only partitions (or questions) allowed are those that correspond to some comparison. There may exist 'crucial' partitions (or questions) that do not correspond to any comparison and consequently are not possible as was shown in the paragraph following (1.10).

At an arbitrary point in the auxiliary problem we suppose that the true state of nature is among m states, $2 \leq m \leq N$. In this unrestricted set-up a procedure is determined by knowing $y \equiv y(m)$, the size of the set to be chosen for the next question. Hence the procedure $T_m^{(M)}$ and also $G(m)$ can be found by the recursive formula

$$(1.24) \qquad G(m) = 1 + \min_{1 \leq y \leq m-1} (\operatorname{Max}(G(y), G(m-y)))$$

with the boundary condition

$$(1.25) \qquad G(1) = 0.$$

Let $\Delta_m^{(M)}$ be the closed interval we select any y in $A^{(M)}$. It is easy to see that $(\ )$ is never empty and consists of a single integer if and only if $\ $ is a power of 2. The expected number of questions under $T_m^{(M)}$ is $H(m)$ given by (1.).

Remark 2. The auxiliary problem and also the expectation problem in remark 1 are relatively easy problems because after each question we are left with a situation entirely similar to the original one. It should be noted that there are no restrictions on the size or nature of the set that we can choose for the question. This is unlike the problem stated in section 1.1, say for $t = n - 1$, where the only partitions (or questions) allowed are those that correspond to some comparison. There may exist "crucial" partitions (or questions) that do not correspond to any comparison and consequently are not possible as was shown in the paragraph following (1.10).

At an arbitrary point in the auxiliary problem we suppose that the true state of nature is among m states, $2 \leq m \leq M$. In this unrestricted set-up a procedure is determined by knowing $y = y(m)$, the size of the set to be chosen for the next question. Hence the procedure $T_m^{(M)}$ and also $G(m)$ can be found by the inductive formula

$$(1.24) \qquad G(m) = 1 + \min_{1 \leq y \leq m-1} \left(\max(G(y), G(m-y)) \right)$$

with the boundary condition

$$(1.25) \qquad G(1) = 0.$$

Let $I^{(M)}$ be the closed interval

$$(1.26) \qquad \Delta_m^{(M)} = [m - 2^{\{\log \frac{m}{2}\}}, 2^{\{\log \frac{m}{2}\}}]$$

which is never void and reduces to a single integer if and only if m is a power of 2.

<u>Lemma 3</u>. The solution of (1.24) and (1.25) is given by

(a) $G(m) = s_m = \{\log m\} \qquad m = 1, 2, \ldots.$

(b) An integer y minimizes the RHS of (1.24) if and only if y belongs to $\Delta_m^{(M)}$, defined in (1.26).

Proof. Trivially $\{\log m\}$ satisfies (1.25). To prove the rest of the lemma we consider the case $y \leq m - y$ or $y \leq \frac{m}{2}$; later we interchange y and $m - y$, if necessary, to get the results for the case $y \geq m - y$. Since $y \leq m - y$

$$\operatorname{Max}(\{\log y\}, \{\log(m-y)\}) = \{\log(m-y)\}$$

and

$$(1.27) \qquad \operatorname*{Min}_{1 \leq y \leq \frac{m}{2}} \{\log(m-y)\} = \{\log\{\tfrac{m}{2}\}\} = \{\log \tfrac{m}{2}\} = \{\log m\} - 1.$$

It follows that (1.24) is satisfied for $1 \leq y \leq \frac{m}{2}$. A similar argument finishes the proof of (a).

To prove (b) for this case it suffices to find all integers y with $1 \leq y \leq \frac{m}{2}$ such that

$$(1.28) \qquad \{\log(m-y)\} = \{\log \tfrac{m}{2}\}.$$

It is easily verified that (1.28) is satisfied if and only if

$$(1.29) \qquad m - 2^{\{\log \frac{m}{2}\}} \leq y \leq \frac{m}{2}.$$

Changing y to $m - y$ for the alternative case we find that

$$(1.25)\qquad \gamma^{(M)} = [m - 2^{\{\log \frac{m}{2}\}},\ 2^{\{\log \frac{m}{2}\}}]$$

which is never void and reduces to a single integer if and only if m is a power of 2.

Lemma 5. The solution of (1.24) and (1.2?) is given by

(a) $G(m) = \beta_m = \{\log m\}$ $m = 1, 2, \ldots$

(b) An integer y minimizes the RHS of (1.24) if and only if y belongs to $\gamma^{(M)}$, defined in (1.25).

Proof. Trivially $\{\log 1\}$ satisfies (1.2?). To prove the rest of the lemma we consider the case $y \le m - y$ or $y \le \frac{m}{2}$; later we interchange y and $m - y$, if necessary, to get the results for the case $y \ge m - y$. Since $y \le m - y$

$$\mathrm{Max}(\{\log y\}, \{\log(m-y)\}) = \{\log(m-y)\}$$

and

$$(1.27)\qquad \min_{1 \le y \le \frac{m}{2}} \{\log(m-y)\} = \{\log \lceil \tfrac{m}{2} \rceil\} = \{\log \tfrac{m}{2}\} = \{\log m\} - 1.$$

It follows that (1.24) is minimized for $1 \le y \le \frac{m}{2}$. A similar argument finishes the proof of (a).

To prove (b) for this case it suffices to find all integers y with $1 \le y \le \frac{m}{2}$ such that

$$(1.28)\qquad \{\log(m-y)\} = \{\log \tfrac{m}{2}\}.$$

It is easily verified that (1.28) is satisfied if and only if

$$(1.29)\qquad m - 2^{\{\log \frac{m}{2}\}} \le y \le \frac{m}{2}.$$

Changing y to $m - y$ for the alternative case we find that

(1.30) $\frac{m}{2} \leq y \leq 2^{\{\log \frac{m}{2}\}}$.

Inequalities (1.29) and (1.30) finish the proof of (b). It should be noted that $\Delta_m^{(M)}$ always contains the closest integer (or integers) to $\frac{m}{2}$.

1.4. Relation between the E-noiseless and M-noiseless properties.

A procedure may be M-noiseless and have E-noise > 0. For example, the R_{FJ} procedure [5] for $n = 6$ is M-noiseless since its maximum number of comparisons $\{H(6!)\} = 10$ and has E-noise equal to $A(6|R_{FJ}) - H(6!) = (9 + \frac{3}{5}) - 19 + \frac{26}{45}) = \frac{1}{45} > 0$ (see Table 2 in Section 3.4 below). For $n = 6$ Picard has given a procedure [11-page 116] which is both M-noiseless and E-noiseless.

The following theorem characterizes procedures that are both E-noiseless and M-noiseless. Let $\mu(n|R)$ be the minimum number of comparisons needed for ranking n items under a procedure R. The range $\rho(n|R)$ of a procedure is defined to be

$$(1.31) \qquad \rho(n|R) = M(n|R) - \mu(n|R).$$

It is shown in [19-Theorem 2] that for $n \geq 2$

$$(1.32) \qquad \rho(n|R) \geq 1$$

for any procedure R that ranks n items.

THEOREM 1. A procedure R for ranking n items is both E-noiseless and M-noiseless if and only if it has minimum range (i.e., if and only if $\rho(n|R) = 1$).

Proof. Let a_j denote the number of branches of length $k - j$ in a tree representing a procedure R, where $k = M(n|R)$. It is

$$\frac{n}{2} \leq \lambda \leq 2^{[\log \frac{n}{2}]}. \tag{1.30}$$

Inequalities (1.29) and (1.30) eliminate proof of (a). It should be noted that $_{(M)}$ always contains the closest integer (or integers) to $\frac{n}{2}$.

1.4. Relation between the E-noiseless and M-noiseless properties.

A procedure may be M-noiseless and have E-noise > 0. For example, the K_{B1} procedure [4] for $n = 6$ is M-noiseless since its minimum number of comparisons $\{H(\Omega)\} = 10$ and has E-noise equal to $A(\Omega|K_{B1}) - H(\Omega) = (9 + \frac{[illegible]}{[illegible]}) - (9 + \frac{25}{[illegible]}) = \frac{[illegible]}{4} > 0$ (see Table in section [illegible] below). For $n = 5$ Picard has given a procedure [11-page 11] which is both M-noiseless and E-noiseless.

The following theorem characterizes procedures that are both E-noiseless and M-noiseless. Let $\lambda(n|K)$ be the minimum number of comparisons needed for ranking n items under a procedure K. The range $\rho(n|K)$ of a procedure is defined to be

$$\rho(n|K) = M(n|K) - \lambda(n|K). \tag{1.31}$$

It is shown in [13-Theorem 2] that for $n \geq 2$

$$\rho(n|K) \geq 1 \tag{1.32}$$

for any procedure K that ranks n items.

THEOREM 1. A procedure K for ranking n items is both E-noiseless and M-noiseless if and only if it has minimum range (i.e., if and only if $\rho(n|K) = 1$).

Proof. Let a_i denote the number of branches of length $k - i$ in a tree representing a procedure K, where $k = M(n|K)$. It is easily seem that we need at least $n - 1$ comparisons so that $0 < a_0 < n!$, $a_j \geq 0$ for $j \geq 1$ and $a_j = 0$ for $j > k - n + 1$. Steinhaus also points out in [19] that the total number of branches is

$$\sum_{j=0}^{k-n+1} a_j = n! , \tag{1.33}$$

and (as can be seen by increasing every branch to length k) we also have

$$\sum_{j=0}^{k-n+1} 2^j a_j = 2^k. \tag{1.34}$$

Also, by definition, the expected length is

$$A(n|R) = \frac{1}{n!} \sum_{j=0}^{k-n+1} (k-j)a_j = k - \frac{1}{n!} \sum_{j=1}^{k-n+1} ja_j. \tag{1.35}$$

Finding a_1 from (1.33) and (1.34) and substituting in (1.35) we get

$$A(n|R) = k - \frac{2^k - n!}{n!} + \frac{1}{n!} \sum_{j=2}^{k-n+1} (2^j - j - 1)a_j . \tag{1.36}$$

If the procedure R is M-noiseless then $k = \{\log n!\}$ and by (1.5) we have $k - \frac{2^k - n!}{n!} = H(n!)$. Since R is also E-noiseless then $A(n|R) = H(n!)$ and hence from (1.36) we get

$$\sum_{j=2}^{k-n+1} (2^j - j - 1)a_j = 0. \tag{1.37}$$

Since the coefficients are non-negative it follows that $a_j = 0$ for $j \geq 2$, i.e., the range of R is one.

Conversely if $a_j = 0$ for $j \geq 2$ then from (1.36) and (1.9) we have easily seen that we need at least $n - 1$ comparisons so that $0 < a_0 < n!$, $a_j \geq 0$ for $j \geq 1$ and $a_j = 0$ for $j > k - n + 1$. Steinhaus also points out in [1] that the total number of branches is

$$\sum_{j=0}^{k-n+1} a_j = n! \quad (1.3)$$

and (as can be seen by increasing every branch to length k) we also have

$$\sum_{j=0}^{k-n+1} 2^{j} a_j = 2^k . \quad (1.4)$$

Also, by definition, the expected length is

$$A(n|R) = \sum_{j=0}^{k-n+1} \frac{a_j}{n!}(k-j) = k - \frac{1}{n!}\sum_{j=1}^{k-n+1} j a_j . \quad (1.5)$$

Eliminating a_1 from (1.3) and (1.4) and substituting in (1.5) we get

$$A(n|R) = k - \frac{2^k - n!}{n!} + \frac{1}{n!}\sum_{j=2}^{k-n+1} (2^j - j - 1) a_j . \quad (1.6)$$

If the procedure R is M-costless then $k = \lceil \log n! \rceil$ and by (1.[illegible]) we have $k - \frac{2^k - n!}{n!} = H(n!)$. Since R is also A-costless then $A(n|R) = H(n!)$ and hence from (1.6) we get

$$\sum_{j=2}^{k-n+1} (2^j - j - 1) a_j = 0 . \quad (1.7)$$

Since the coefficients are non-negative it follows that $a_j = 0$ for $j \geq 2$, i.e., the range of R is one.

Conversely if $a_j = 0$ for $j \geq 2$ then from (1.6) and (1.[illegible]) we have

(1.38) $$H(n!) \leq A(n|R) = k - \frac{2^k - n!}{n!}.$$

Since by (1.18) $k \geq \{\log n!\} = s$ (say), let $k = s + i$ where i is a non-negative integer. To prove that any procedure R satisfying (1.38) is both M-noiseless and E-noiseless, it suffices to show that $i = 0$ and equality holds in (1.38). To see this we need only to show that

(1.39) $$s + i - \frac{2^{s+i} - n!}{n!} \leq s - \frac{2^s - n!}{n!} = H(n!)$$

and equality holds if and only if $i = 0$. The inequality in (1.39) is equivalent to

(1.40) $$n!i \leq 2^s(2^i - 1).$$

Since $i \leq 2^i - 1$ for all i and $n! \leq 2^s$ for all n the inequality (1.40) holds. Clearly equality holds in (1.40) if and only if $i = 0$, and this proves our theorem.

To prove a relation between E-noiseless and M-noiseless procedures we need

<u>Lemma 4</u>. For each m $(m = 2, 3, \ldots)$

(1.41) $$\Delta_m^{(E)} \subset \Delta_m^{(M)}$$

where $\Delta_m^{(E)}$ and $\Delta_m^{(M)}$ are given by (1.23) and (1.26), respectively.

Proof. Let $y \in \Delta_m^{(E)}$. Clearly

(1.42) $$\mathrm{Min}(y, m-y) \leq \frac{m}{2} \leq \mathrm{Max}(y, m-y).$$

On the other hand $2^{\{\log \frac{m}{2}\}-1}$ and $2^{\{\log \frac{m}{2}\}}$ are the largest and smallest powers of 2, respectively, such that

(1.38) $N(n!) \leq A(n|K) = k - \frac{2^k - n!}{n!}$.

Since by (1.18) $k \geq \lceil \log n! \rceil = s$ (say). Let $k = s + t$ where t is a non-negative integer. To prove that any procedure K satisfying (1.38) is both M-noiseless and C-noiseless, it suffices to show that $t = 0$ and equality holds in (1.38). To see this we need only to show that

(1.39) $$s + t - \frac{2^{s+t} - n!}{n!} \leq s - \frac{2^s - n!}{n!} = H(n!)$$

and equality holds if and only if $t = 0$. The inequality in (1.39) is equivalent to

(1.40) $$n!\,t \leq 2^s(2^t - 1).$$

Since $t \leq 2^t - 1$ for all t and $n! \leq 2^s$ for all n the inequality (1.40) holds. Clearly equality holds in (1.40) if and only if $t = 0$, and this proves our theorem.

To prove the relation between d-noiseless and M-noiseless procedures we need

Lemma 4. For each m ($m = 2, 3, \ldots$)

(1.41) $$A^m_{(d)} = A^m_{(M)}$$

where $A^m_{(d)}$ and $A^m_{(M)}$ are given by (1.21) and (1.24), respectively.

Proof. Let $y \in A^m_{(d)}$. Clearly

(1.42) $$\mathrm{Min}(y, m-y) \leq \frac{m}{2} \leq \mathrm{Max}(y, m-y).$$

On the other hand $2^{\lceil \log \frac{m}{2} \rceil - 1}$ and $2^{\lceil \log \frac{m}{2} \rceil}$ are the largest and smallest powers of 2, respectively, such that

$$(1.43) \qquad 2^{\{\log \frac{m}{2}\}-1} \leq \frac{m}{2} \leq 2^{\{\log \frac{m}{2}\}}.$$

To satisfy the condition that there is no power of 2 between y and $m - y$, it is necessary and sufficient that

$$(1.44) \qquad 2^{\{\log \frac{m}{2}\}-1} \leq \mathrm{Min}(y,\ m-y) \leq \mathrm{Max}(y,\ m-y) \leq 2^{\{\log \frac{m}{2}\}}.$$

The inequalities in (1.44) implies in particular that

$$(1.45) \qquad m - 2^{\{\log \frac{m}{2}\}} \leq y \leq 2^{\{\log \frac{m}{2}\}}$$

i.e., that $y \in \Delta_m^{(M)}$, and this proves our lemma.

<u>THEOREM 2</u>. If a procedure is E-noiseless it is also M-noiseless.

Proof. Consider any non-terminal point reached in the course of carrying out the comparisons and suppose that we now have m possible states of nature (or cases), one of which is the true ordering of the n items. At the outset $m = n!$. Let $y = y(m;\ x_i,\ x_j)$ be the size of the subset that contains the true ordering as a result of the comparison x_i vs. x_j. Using Lemma (2) in [16 section 6] we conclude that a procedure is E-noiseless if and only if $y \in \Delta_m^{(E)}$. Similarly from Lemma 3 we conclude that a procedure is M-noiseless if and only if $y \in \Delta_m^{(M)}$. Hence Lemma 4 completes the proof of our theorem.

Remark 1. Note that the converse of Theorem 2, which is one of the questions raised in [10 - page 47], is not necessarily true. The Ford and Johnson procedure, [5] for $n = 6$ given in the beginning of this section, provides a counter-example.

Remark 2. Theorem 2 can be used in searching for an E-noiseless procedure, since it shows that we can restrict our attention to M-noiseless procedures.

$$2^{[\log \frac{m}{2}]-1} \le \frac{n}{2} \le 2^{[\log \frac{m}{2}]} . \tag{4.43}$$

To satisfy the condition that there is no power of 2 between y and $n - y$, it is necessary and sufficient that

$$2^{[\log \frac{m}{2}]-1} \le \min(y, n-y) \le \max(y, n-y) \le 2^{[\log \frac{m}{2}]} . \tag{4.44}$$

The inequalities in (4.44) implies in particular that

$$n - 2^{[\log \frac{m}{2}]} \le y \le 2^{[\log \frac{m}{2}]} \tag{4.45}$$

i.e., that $y \in A_n^{(M)}$, and this proves our Lemma.

THEOREM 2. If a procedure is H-noiseless it is also M-noiseless.

Proof. Consider any non-terminal point reached in the course of carrying out the comparisons and suppose that we now have n possible states of nature (or cases), one of which is the true ordering of the n items. At the outset $n = n!$. Let $y = h(n; x_i, x_j)$ be the size of the subset that contains the true ordering as a result of the comparison x_i vs. x_j. Using Lemma (2) in [10 section 6] we conclude that a procedure is H-noiseless if and only if $y \in A_n^{(H)}$. Similarly from Lemma 3 we conclude that a procedure is M-noiseless if and only if $y \in A_n^{(M)}$. Hence Lemma 4 completes the proof of our theorem.

Remark 1. Note that the converse of Theorem 2, which is one of the questions raised in [10—page 8], is not necessarily true. The Ford and Johnson procedure, [1] for $n = 6$ given in the beginning of this section, provides a counter-example.

Remark 2. Theorem 2 can be used in searching for an H-noiseless procedure, since it shows that we can restrict our attention to M-noiseless procedures.

Now we can combine Theorems 1 and 2 to strengthen them as follows:

THEOREM 1a. A procedure for ranking n items is E-noiseless if and only if it has minimum range (i.e., if and only if the procedure is of range 1).

THEOREM 2a. If a procedure is E-noiseless it is M-noiseless, and only M-noiseless procedures of range 1 are E-noiseless.

1.5. Subclasses of procedures.

In previous sections we introduced two different optimality criteria and mentioned that existing procedures are 'optimal' only for certain values of n. The idea behind introducing subclasses is that we may be able to find an M-optimal (or E-optimal) procedure in a subclass when it is difficult to find an M-optimal (or E-optimal) procedure in $\mathcal{R}_t$. This idea turns out to be helpful since we may be able to find lower bounds for the subclass when lower bounds for the whole class are difficult to obtain. We may also be able to use a systematic method to find an 'optimal' procedure for the subclass whereas it seems to be a difficult task to invent a method for finding the 'optimal' procedure in the unrestricted class $\mathcal{R}_t$. First we introduce some terminology and definitions.

We recall that if $n = 2^r$ then a knock-out tournament finds the largest item among $x_1, x_2, \ldots, x_n$ by making 2^r-1 comparisons as follows: 2^{r-1} comparisons in the first round x_1 vs. $x_2, \ldots, x_{n-1}$ vs. x_n; 2^{r-2} comparisons in the second round pairing off the winners (larger items) of the first round and so on until the winner is found. Without loss of any generality it is possible to assume that the winners of the first round are those with even indices, the winners of the second

Now we can combine Theorems 1 and 2 to strengthen them as follows:

THEOREM 1A. A procedure for ranking n items is N-noiseless if and only if it has minimum range (i.e., if and only if the procedure is of range 1).

THEOREM 2A. If a procedure is N-noiseless it is M-noiseless, and only M-noiseless procedures of range 1 are N-noiseless.

1.5. Subclasses of Procedures.

In previous sections we introduced two different optimality criteria and mentioned that existing procedures are 'optimal' only for certain values of n. The idea behind introducing subclasses is that we may be able to find an M-optimal (or N-optimal) procedure in a subclass when it is difficult to find an M-optimal (or N-optimal) procedure in $\mathcal{P}_n$. This idea turns out to be helpful since we may be able to find lower bounds for the subclass when lower bounds for the whole class are difficult to obtain. We may also be able to use a systematic method to find an 'optimal' procedure for the subclass whereas it seems to be a difficult task to invent a method for finding the 'optimal' procedure in the unrestricted class $\mathcal{P}_n$. First we introduce some terminology and definitions.

We recall that if $n = 2^k$ then a knock-out tournament finds the largest item among $x_1, x_2, \ldots, x_n$ by making 2^k-1 comparisons as follows: 2^{k-1} comparisons in the first round: x_1 vs. $x_2, \ldots, x_{n-1}$ vs. x_n; 2^{k-2} comparisons in the second round pair up all the winners (larger items) of the first round and so on until the winner is found. Without loss of any generality it is possible to assume that the winners of the first round are those with even indices; the winners of the second round are those with indices that are multiples of 4, etc. This notation is convenient.

<u>Complete pairing</u>: Let $n = 2^{r_1} + 2^{r_2} + \ldots + 2^{r_\alpha}$ be the binary expansion of n, i.e., 2^{r_1} is the highest power of 2 in n and 2^{r_j} is the highest power of 2 in $n - 2^{r_1} - \ldots - 2^{r_{j-1}}$ where $2 \leq j \leq \alpha$. The complete pairing consists of $\sum_{j=1}^{\alpha} (2^{r_j} - 1)$ comparisons, consisting of a knock-out tournament for each subset of size 2^{r_j} $(j = 1, 2, \ldots, \alpha)$. Without loss generality we can take $x_1, x_2, \ldots, x_{\beta_1}$ for the first 'group', where $\beta_1 = 2^{r_1}$; and for $j \geq 2$ we can take $x_{i+1}, x_{i+2}, \ldots, x_{i+\beta_j}$ for the j^{th} 'group', where $i = i(j) = 2^{r_1} + \ldots + 2^{r_{j-1}}$ and $\beta_j = 2^{r_j}$. After the complete pairing of n items we are left with α configurations; for example, for $n = 13 = 2^3 + 2^2 + 2^0$ we have the 3 configurations given in Fig. 3.

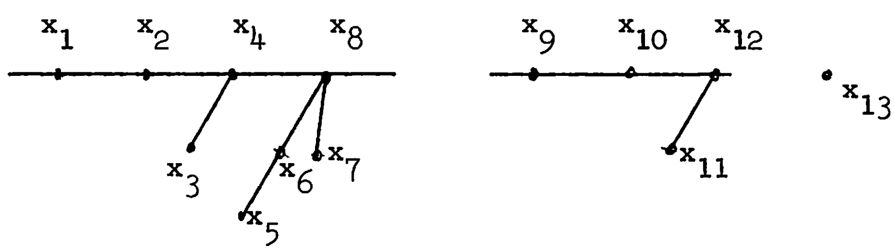


Fig. 3. <u>Complete pairing of 13 items</u>.

The number of comparisons for complete pairing of n items $p = p(n)$ as given by the lemma in [16 Section 4] is

$$(1.46) \qquad p = n - \alpha = \sum_{j=1}^{[\log n]} \left[\frac{n}{2^j}\right] = \pi(n!)$$

where α is the number of 1's in the binary expansion of n and $\pi = \pi(i)$ is the largest integer such that 2^π divides i, i.e., π and $c = c(i)$ are non-negative integers defined by

$$(1.47) \qquad i = 2^\pi(2c+1).$$

round are those with indices that are multiples of 4, etc. This notation is convenient.

Complete pairing: Let $n = 2^{r_1} + 2^{r_2} + \dots + 2^{r_w}$ be the binary expansion of n, i.e., 2^{r_1} is the highest power of 2 in n and 2^{r_j} is the highest power of 2 in $n - 2^{r_1} - \dots - 2^{r_{j-1}}$ where $2 \le j \le w$. The complete pairing consists of $\sum_{j=1}^{w} (2^{r_j}-1)$ comparisons, consisting of a knock-out tournament for each subset of size 2^{r_j} ($j = 1, 2, \dots, w$). Without loss generality we can take $x_1, x_2, \dots, x_{k_1}$ for the first 'group', where $k_1 = 2^{r_1}$; and for $j \ge 2$ we can take $x_{k+1}, x_{k+2}, \dots, x_{k+k_j}$ for the j^{th} 'group', where $k = k(j) = 2^{r_1} + \dots + 2^{r_{j-1}}$ and $k_j = 2^{r_j}$. After the complete pairing of n items we are left with w conditions; for example, for $n = 13 = 2^3 + 2^2 + 2^0$ we have the 3 conditions given in Fig. 1.

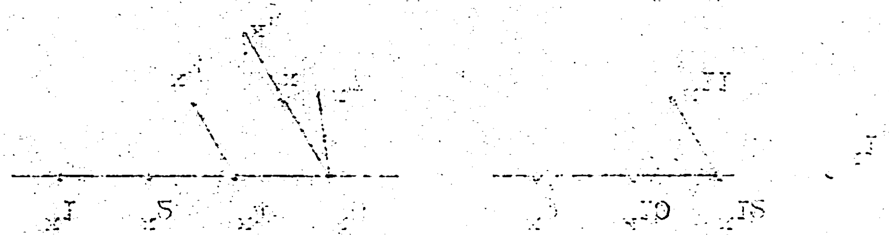


Fig. 1. Complete pairing of 13 items.

The number of comparisons for complete pairing of n items $p = p(n)$ as given by the lemma in [10 Section 4] is

$$(1.45) \qquad p = n - w = \sum_{j=1}^{[\log_2 n]} \left[\frac{n}{2^j}\right] = r(n!)$$

where w is the number of 1's in the binary expansion of n and $r = r(k)$ is the largest integer such that 2^r divides k, i.e., r and $c = c(k)$ are non-negative integers defined by

$$(1.46) \qquad k = 2^r(2c+1).$$

Ordinary pairing of n items consists simply of $[\frac{n}{2}]$ comparisons x_1 vs. x_2, x_3 vs. x_4,....

One may be interested in an E-optimal (M-optimal) procedure in a subclass $\mathcal{C}_t$ of $\mathcal{R}_t$ for ranking the t largest among n items. Let

$$(1.48) \qquad L(t, n|\mathcal{C}_t) = \underset{C\epsilon\mathcal{C}_t}{\text{Min}} \; A_t(n|C)$$

and

$$(1.49) \qquad L_{\text{Max}}(t, n|\mathcal{C}_t) = \underset{C\epsilon\mathcal{C}_t}{\text{Min}} \; M_t(n|C).$$

Obviously

$$(1.50) \qquad L(t, n) \leq L(t, n|\mathcal{C}_t)$$

and

$$(1.51) \qquad L_{\text{Max}}(t, n) \leq L_{\text{Max}}(t, n|\mathcal{C}_t);$$

in particular if $\mathcal{C}_t = \mathcal{R}_t$ we have equality in (1.50) and in (1.51). It is interesting to note that we may have equality in (1.50) (in (1.51)) which would imply that to find an E-optimal (M-optimal) procedure among all procedures one need only find an E-optimal (M-optimal) procedure in class $\mathcal{C}_t$. ('Optimality' in a class $\mathcal{C}_t$ is defined in a similar fashion as 'optimality' in $\mathcal{R}_t$, i.e., RHS's of (1.2) and (1.14) are replaced by $L(t, n|\mathcal{C}_t)$ and $L_{\text{Max}}(t, n|\mathcal{C}_t)$ respectively.) More precisely a class $\mathcal{S}_t$ is called E-complete (M-complete) if

$$(1.52) \qquad L(t, n|\mathcal{S}_t) = L(t, n)$$

$$((1.53) \qquad L_{\text{Max}}(t, n|\mathcal{S}_t) = L_{\text{Max}}(t, n)).$$

Ordinary pairing of n items consists simply of $[\frac{n}{2}]$ comparisons x_1 vs. x_2, x_3 vs. x_4, ...

One may be interested in an R-optimal (M-optimal) procedure in a subclass $\mathcal{C}_s$ of $\mathcal{D}_s$ for ranking the t largest among n items. Let

$$(1.48)\qquad L(t, n|\mathcal{C}_s) = \underset{C \in \mathcal{C}_s}{\mathrm{Min}}\ L_s(n;C)$$

and

$$(1.49)\qquad L_{Max}(t, n|\mathcal{C}_s) = \underset{C \in \mathcal{C}_s}{\mathrm{Min}}\ M_s(n;C).$$

Obviously

$$(1.50)\qquad L(t, n) \le L(t, n|\mathcal{C}_s)$$

and

$$(1.51)\qquad L_{Max}(t, n) \le L_{Max}(t, n|\mathcal{C}_s);$$

in particular if $\mathcal{C}_s = \mathcal{D}_s$ we have equality in (1.50) and in (1.51). It is interesting to note that we may have equality in (1.50) (in (1.51)) which would imply that to find an R-optimal (M-optimal) procedure among all procedures one need only find an R-optimal (M-optimal) procedure in class $\mathcal{C}_s$. ('Optimality' in a class $\mathcal{C}_s$ is defined in a similar fashion as 'optimality' in $\mathcal{D}_s$, i.e., RHS's of (1.2) and (1.14) are replaced by $L(t, n|\mathcal{C}_s)$ and $L_{Max}(t, n|\mathcal{C}_s)$ respectively.) More precisely a class $\mathcal{C}_s$ is called R-sufficient (M-sufficient) if

$$(1.52)\qquad L(t, n|\mathcal{C}_s) = L(t, n)$$

$$((1.53)\qquad L_{Max}(t, n|\mathcal{C}_s) = L_{Max}(t, n)).$$

The class $\mathscr{R}_t$ is trivially both E- and M-complete.

Important subclasses that we consider are

<u>The Inductive class</u> $\mathscr{I}_t$: This class consists of all procedures that require, for fixed t and each n, ($n = 2, 3, \ldots$), ranking of the t largest of $n - 1$ items inductively. For the special case $t = n - 1$ we denote $\mathscr{I}_{n-1}$ by $\mathscr{I}$. The class $\mathscr{I}$ is studied in Chapter 2.

<u>The Semi-inductive class</u> $\mathscr{J}_t$: This class consists of all procedures such as $R_{n,t}$ that require, for fixed t and each n ($n = 2, 3, \ldots$), the following 2 steps:

1. Ordinary pairing of n items, which as a result there will be $[\frac{n}{2}]$ larger items.

2. Ranking the t largest of $[\frac{n}{2}]$ larger items in step 1 according to $R_{[\frac{n}{2}],t}$.

For the special case $t = n - 1$ we denote $\mathscr{J}_{n-1}$ by $\mathscr{J}$. An example of procedures in $\mathscr{J}$ is the procedure given in [5]. We study the class $\mathscr{J}$ in Chapter 3.

The <u>complete pairing class</u> $\mathscr{P}_t$: This class consists of all procedures that initially carry out the complete pairing on the n items. It is a proper subclass of $\mathscr{R}_t$.

It is easy to show that $\mathscr{P}_t$ is an E- and M-sufficient class for $t = 1$. Let $\mathscr{P}_{n-1} \equiv \mathscr{P}$. We conjecture that $\mathscr{P}$ is an E- and M-complete class for ranking n items. We prove a somewhat weaker result in Lemma 5 below.

We may be interested in finding a procedure for ranking n items given a partial pre-existing order among n items. Suppose there

The class $\mathcal{A}_t$ is trivially both E- and M-sufficient.

Important subclasses that we consider are

<u>The inductive class</u> $\mathcal{I}_t$: This class consists of all procedures that require, for fixed t and each n, $(n = 2, 3, \ldots)$, ranking of the t largest of $n - 1$ items inductively. For the special case $t = n - 1$ we denote $\mathcal{I}_{n-1}$ by $\mathcal{I}$. The class $\mathcal{I}$ is studied in Chapter 2.

<u>The semi-inductive class</u> $\mathcal{S}_t$: This class consists of all procedures such as $R_{n,t}$ that require, for fixed t and each n $(n = 2, 3, \ldots)$, the following 2 steps:

1. Ordinary pairing of n items, which as a result there will be $[\frac{n}{2}]$ larger items.

2. Ranking the t largest of $[\frac{n}{2}]$ larger items in step 1 according to $R_{[\frac{n}{2}],t}$.

For the special case $t = n - 1$ we denote $\mathcal{S}_{n-1}$ by $\mathcal{S}^*$. An example of procedures in $\mathcal{S}$ is the procedure given in [ ]. We study the class $\mathcal{S}$ in Chapter 3.

<u>The complete pairing class</u> $\mathcal{P}_t$: This class consists of all procedures that initially carry out the complete pairing on the n items. It is a proper subclass of $\mathcal{S}_t$.

It is easy to show that $\mathcal{P}_t$ is an E- and M-sufficient class for $t = 1$. Let $\mathcal{P}_{n-1} = \mathcal{P}$. We conjecture that $\mathcal{P}$ is an E- and M-sufficient class for ranking n items. We prove a somewhat weaker result in Lemma below.

We may be interested in finding a procedure for ranking n items given a partial pre-existing order among n items. Suppose there are m permutations of n items consistent with the pre-existing order. A procedure R for ranking n items is said to be <u>conditionally</u> <u>E-noiseless</u> (<u>conditionally</u> <u>M-noiseless</u>) given a pre-existing order among items if the expected number of comparisons (maximum number of comparisons) under R is $H(m)$ ($\{H(m)\}$).

<u>Lemma 5</u>. If a procedure R is conditionally E-noiseless (conditionally M-noiseless) given the complete pairing on n items then a procedure R' in $\mathcal{P}$ and identical to R after the complete pairing is E-noiseless (M-noiseless) among all possible procedures.

Proof. We prove that the procedure R' is E-noiseless, the parenthetical case can be shown similarly. Since there are $m = 2c(n!) + 1$ permutations consistent with the complete pairing of n items, where $c(i)$ is defined by (1.47); and since R is conditionally E-noiseless then the expected number of comparisons under R after the complete pairing is $H(2c(n!) + 1)$. Furthermore the procedure R' requires $\pi(n!)$ additional comparisons where $\pi(i)$ is defined by (1.47). Hence

$$A(n|R') = \pi(n!) + H(2c(n!) + 1). \tag{1.54}$$

To prove that R' is an E-noiseless procedure among all possible procedures we need to prove

$$H(n!) = \pi(n!) + H(2c(n!) + 1). \tag{1.55}$$

To show (1.55) we note that by (1.5) and (1.47) with $i = n!$

$$H(n!) = \{\log n!\} - \frac{2^{\{\log n!\}} - n!}{n!}$$

are $n!$ permutations of n items consistent with the pre-existing order. A procedure R for ranking n items is said to be conditionally H-noiseless (conditionally M-noiseless) given a pre-existing order among items if the expected number of comparisons (maximum number of comparisons) under R is $H(m)$ ($\lceil H(m)\rceil$).

Lemma 1. If a procedure R is conditionally H-noiseless (conditionally M-noiseless) given the complete pairing of n items then procedure R' in $\mathcal{P}$ and identical to R after the complete pairing is H-noiseless (M-noiseless) among all possible procedures.

Proof. We prove that the procedure R' is H-noiseless, the parenthetical case can be shown similarly. Since there are $m = 2e(n!) + 1$ permutations consistent with the complete pairing of n items, where $e(i)$ is defined by (1.47); and since R is conditionally noiseless then the expected number of comparisons under R after the complete pairing is $H(2e(n!) + 1)$. Furthermore the procedure R' requires $r(n!)$ additional comparisons where $r(\,)$ is defined by (1.47). Hence

$$(1.54) \qquad A(n|R') = r(n!) + H(2e(n!) + 1).$$

To prove that R' is an H-noiseless procedure among all possible procedures we need to prove

$$(1.55) \qquad H(n!) = r(n!) + H(2e(n!) + 1).$$

To show (1.55) we note that by (1.5) and (1.47) with $k = n!$

$$H(n!) = \lceil \log n! \rceil - \frac{2^{\lceil \log n! \rceil} - n!}{n!}$$

$$= \pi(n!) + \{\log(2c(n!)+1)\} - \left(\frac{2^{\{\log(2c(n!)+1)\}} - (2c(n!)+1)}{2^{\pi(n!)}(2c(n!)+1)}\right)2^{\pi(n!)}$$

$$= \pi(n!) + H(2c(n!) + 1) \qquad \text{qed.}$$

Sobel [16-Section 2] has shown that for $t = 2$ and $n \neq 2^r$ the class $\mathcal{P}_t$ may not be an E-complete class. He has given a procedure R_E that has expectation $6 + \frac{1}{2}$ for $n = 6$, which is smaller than $6 + \frac{2}{3}$ the expectation of an E-optimal procedure in $\mathcal{P}_2$. An E-optimal procedure in $\mathcal{P}_2$ and $n = 6$ can be found by considering all possible continuations after the complete pairing.

$$= r(n!) + [\log_2(2c(n!)+1)] - \left(\frac{2^{[\log_2(2c(n!)+1)]}-(2c(n!)+1)}{2^{r(n!)}(2c(n!)+1)}\right)2^{r(n!)}$$

$$= r(n!) + H(2c(n!) + 1) \qquad \text{qed.}$$

Sobel [16-Section 2] has shown that for $t = 2$ and $n \leqq 2^r$ the class $\mathcal{S}_t$ may not be an R-sufficient class. He has given a procedure R_* that the expectation $5 + \frac{1}{2}$ for $n = 6$, which is smaller than $5 + \frac{2}{3}$ the expectation of an R-optimal procedure in $\mathcal{S}_2$. An R-optimal procedure in $\mathcal{S}_2$ and $n = 6$ can be found by considering all possible continuations after the complete pattern.

CHAPTER II

On the Inductive Class $\mathcal{J}$

2.1. Introduction.

Here we consider in detail the problem formulated in 1.1 for the special case $t = n - 1$. Apparently Steinhaus was the first to introduce the problem in [17], with M-optimality criteria, where he heuristically suggested the procedure that we call R_S and explain in the following paragraph. If m items are ordered as $x_1 < x_2 <\dots< x_m$ the rank of x_{m-i+1} is defined to be i; and the median of m items, for even m, is either one of items with rank $\frac{m}{2}$ or $\frac{1}{2}(m+2)$.

For any j, with $2 \leq j \leq n$ we first rank $j - 1$ items inductively. Then find the median of $j - 1$ items already ranked. Now insert the j^{th} item among the $j - 1$ items already ranked by comparing it with the median of the $j - 1$ items; if it is larger then compare it with the median of the 'upper group'; if it is smaller then compare it with the median of the 'lower group'. Continue the process until the proper place of the j^{th} items is found. Let $M(n) \equiv M(n|R_S)$. Then

$$(2.1) \qquad s_n \equiv M(n) - M(n-1) = \{\log n\} = \{H(n)\} = 1 + [\log (n-1)]$$

where $M(1) = 0$.

The quantity $s_n = M(n) - M(n-1)$ is in fact the maximum number of comparisons needed to insert one item among $n - 1$ items already ranked under the R_S procedure. From (2.1) one can obtain as in [4] the Steinhaus result

CHAPTER II

On the Inductive Class

2.1. Introduction.

Here we consider in detail the problem formulated in 1.1 for the special case $t = n - 1$. Apparently Steinhaus was the first to introduce the problem in [17], with M-optimality criteria. Here he heuristically suggested the procedure that we call R_0 and explain in the following paragraph. If m items are ordered as $x_1 < x_2 < \dots < x_m$ the rank of x_{m-i+1} is defined to be i; and the median of m items, for even m, is either one of items with rank $\frac{m}{2}$ or $\frac{1}{2}(m+2)$.

For any j, with $2 \le j \le n$ we first rank $j - 1$ items inductively. Then find the median of $j - 1$ items already ranked. Now insert the j^{th} item among the $j - 1$ items already ranked by comparing it with the median of the $j - 1$ items; if it is larger than compare it with the median of the upper group; if it is smaller than compare it with the median of the lower group. Continue the process until the proper place of the j^{th} item is found. Let $M(n) = M(n;R_0)$. Then

$$s_n = M(n) - M(n-1) = \lceil \log_2 n \rceil = \lceil L(n) \rceil = 1 + [\log_2 (n-1)] \tag{2.1}$$

where $M(1) = 0$.

The quantity $s_n = M(n) - M(n-1)$ is in fact the maximum number of comparisons needed to insert one item among $n - 1$ items already ranked under the R_0 procedure. From (2.1) one can obtain as in [4] the Steinhaus result

$$(2.2) \qquad M(n) = 1 + ns - 2^s$$

where $s = s_n$ is given by (2.1). In [17-1950 edition] Steinhaus conjectures that R_S is an M-optimal procedure but this remark is removed in the 1960 edition. In [18] he gives a procedure for $n = 5$ that requires at most 7 comparisons to rank 5 items, whereas $M(5) = 8$. Hence R_S is not an M-optimal procedure.

Let $A(n) = A(n|R_S)$. It has been noted by several authors, e.g., in [8], [9] and [11], that

$$(2.3) \qquad A(n) - A(n-1) = H(n)$$

where $A(1) = 0$, and hence

$$(2.4) \qquad A(n) = \sum_{j=2}^{n} H(j).$$

Here $A(n) - A(n-1)$ is the expected number of comparisons needed for inserting one item among the $n - 1$ already ranked under the R_S procedure. The procedure given in [18] is the same as R_{FJ} procedure [5] for $n = 5$. It can be shown that $A(5|R_{FJ}) = 6 + \frac{14}{15} < 7 + \frac{1}{15} = A(5)$. Hence the R_S procedure is not E-optimal either.

2.2. <u>The inductive procedure R_I and its expectation.</u>

As is mentioned in Section (2.1) the R_S procedure is neither M-optimal nor E-optimal among all possible procedures. Nevertheless, in this section we prove, among other things, that R_S procedure is both E-optimal and M-optimal in the class $\mathcal{I}$ of inductive procedures.

We now wish to find an inductive procedure R_I, using the technique of dynamic programing, which is E-optimal in $\mathcal{I}$. The method will produce all inductive procedures that are E-optimal in $\mathcal{I}$.

$$M(n) = 1 + ns - 2^s \tag{2.2}$$

where $s = s_n$ is given by (2.1). In [17-1950 edition] Steinhaus conjectures that R_N is an M-optimal procedure but this remark is removed in the 1960 edition. In [10] he gives a procedure for $n = 5$ that requires at most 7 comparisons to rank 5 items, whereas $M(5) = 8$. Hence R_N is not an M-optimal procedure.

Let $A(n) = A(n|R_N)$. It has been noted by several authors, e.g., in [9], [4] and [11], that

$$A(n) - A(n-1) = E(n) \tag{2.3}$$

where $A(1) = 0$, and hence

$$A(n) = \sum_{j=2}^{n} E(j). \tag{2.4}$$

Here $A(n) - A(n-1)$ is the expected number of comparisons needed for inserting one item among the $n - 1$ already ranked under the R_N procedure. The procedure given in [10] is the same as R_{PE} procedure [4] for $n = 5$. It can be shown that $A(5|R_{PE}) = 6 + \frac{14}{15} < 7 + \frac{1}{5} = A(5)$. Hence the R_N procedure is not E-optimal either.

2.2. The inductive procedure R_N and its extension.

As is mentioned in Section (2.1) the R_N procedure is neither M-optimal nor E-optimal among all possible procedures. Nevertheless, in this section we prove, among other things, that R_N procedure is both M-optimal and E-optimal in the class $\mathcal{J}$ of inductive procedures.

We now wish to find an inductive procedure R_E using the technique of dynamic programming, which is E-optimal in $\mathcal{J}$. The method will produce all inductive procedures that are E-optimal in $\mathcal{J}$.

Let $S(n) \equiv A(n|R_I)$ and let

$$\Delta S(n) = S(n) - S(n-1). \tag{2.5}$$

We define R_I for $n \geq 2$ by the recursive formula

$$\Delta S(n) = 1 + \min_{1 \leq y \leq n-1} \left(\frac{y}{n} \Delta S(y) + \frac{n-y}{n} \Delta S(n-y)\right) \tag{2.6}$$

and the boundary conditions

$$S(0) = S(1) = 0. \tag{2.7}$$

The basic idea in using this method of recursion was introduced by Sobel in [14], [15] and also in a previous paper on group testing referred to as reference 7 in [15]. In fact let $n - 1$ items be ranked inductively. Then LHS of (2.6) denotes the minimum additional expected number of comparisons needed to insert the n^{th} item under R_I. To get RHS of (2.6) first we compare the n^{th} item with the one of rank y among the $n - 1$ items already ranked. The probability that the n^{th} item being larger in the first comparison is $\frac{y}{n}$, due to the initial randomness of the n items. Given that the n^{th} item is larger than the item of rank y the expected number of comparisons for finding its proper place among the $y - 1$ items is $\Delta S(y) = S(y) - S(y-1)$. Similarly $\frac{n-y}{n}$ is the probability that the n^{th} item is smaller than the one of rank y. Given the latter event, the expected number of comparisons for finding its proper place among $n - y - 1$ items is $\Delta S(n-y) = S(n-y) - S(n-y-1)$. The integer 1 on RHS of (2.6) represents the 'present' comparison of the n^{th} item and the one of rank y.

Let $g(n) = A(n|R_y)$ and let

$$\Delta S(n) = S(n) - S(n-1). \tag{2.5}$$

We define R_y for $n \geq 2$ by the recursive formula

$$\Delta S(n) = 1 + \min_{1 \leq y \leq n-1} \left(\frac{y}{n} \Delta S(y) + \frac{n-y}{n} \Delta S(n-y) \right) \tag{2.6}$$

and the boundary conditions

$$S(0) = S(1) = 0. \tag{2.7}$$

The basic idea in using this method of recursion was introduced by Sobel in [14], [15] and also in a previous paper on group testing referred to as reference [4] in [15]. In fact let $n - 1$ items be ranked inductively. Then LHS of (2.6) denotes the minimum additional expected number of comparisons needed to insert the n^{th} item under R_y. To get RHS of (2.6) first we compare the n^{th} item with the one of rank y among the $n - 1$ items already ranked. The probability that the n^{th} item being larger in the first comparison is $\frac{y}{n}$, due to the initial randomness of the n items. Given that the n^{th} item is larger than the item in rank y, the expected number of comparisons for finding its proper place among the $y - 1$ items is $\Delta S(y) = S(y) - S(y-1)$. Similarly $\frac{n-y}{n}$ is the probability that the n^{th} item is smaller than the one of rank y. Given the latter event, the expected number of comparisons for finding its proper place among $n - y - 1$ items is $\Delta S(n-y) = S(n-y) - S(n-y-1)$. The integer 1 in RHS of (2.6) represents the 'present' comparison of the n^{th} item and the one of rank y.

The formulas (2.6) and (2.7) define the whole procedure R_I. For $n \geq 2$ at least one y, $1 \leq y \leq n - 1$, is found and the procedure then is defined inductively for all n. Thus the problem of finding an E-optimal procedure in $\mathcal{I}$ is reduced to solving the recursive formula (2.6) with boundary condition (2.7).

If we define for $j \geq 1$

$$(2.8) \qquad h(j) = j(S(j) - S(j-1))$$

then (2.6) can be written in the form: for $n \geq 2$

$$(2.9) \qquad h(n) = n + \min_{1 \leq y \leq n-1} (h(y) + h(n-y))$$

with the boundary condition

$$(2.10) \qquad h(1) = 0.$$

The recursive relation (2.9) is exactly the same as (2.13) in [14], (4.1) in [15] and (6.3) in [16] where it was investigated extensively by Sobel. Here we list some results of this observation.

(a) For any $n \geq 2$ an integer $y = y(n)$ will yield the minimum in (2.9) if and only if there is no power of 2 strictly between y and $n - y$ (see Lemma 2 in Section 6 of [16]). The set of all such y values has already been denoted by $\Delta_n^{(E)}$ in (1.23).

(b) The set $\Delta_n^{(E)}$ always includes the median, say $[\frac{n}{2}]$, of $1, 2, \ldots, n - 1$.

(c) Using the boundary condition (2.10) we obtain

$$(2.11) \qquad h(n) = n(1 + \{\log n\}) - 2^{\{\log n\}}.$$

The formulas (2.7) and (2.8) define the whole procedure P_1. For $n \geq 2$ at least one y, $1 \leq y \leq n - 1$, is found and the procedure then is defined inductively for all n. Thus the problem of finding an H-optimal procedure in $\mathcal{P}$ is reduced to solving the recursive formula (2.8) with boundary condition (2.7).

If we define for $j \geq 1$

$$(2.8) \qquad h(j) = j(s(j) - s(j-1))$$

then (2.5) can be written in the form: for $n \geq 2$

$$(2.9) \qquad h(n) = n + \operatorname*{Min}_{1 \leq y \leq n-1} (h(y) + h(n-y))$$

with the boundary condition

$$(2.10) \qquad h(1) = 0.$$

The recursive relation (2.9) is exactly the same as (2.1) in [15], (4.1) in [1] and (5.1) in [10] where it was investigated extensively by Hobel. Here we list some results of this observation.

(a) For any $n \geq 2$ an integer $y = y(n)$ will yield the minimum in (2.9) if and only if there is no power of 2 strictly between y and $n - y$ (see Lemma 2 in Section 3 in [15]). The set of all such y values has already been denoted by $Y_n^{(1)}$ in (1.2.1).

(b) The set $Y_n^{(1)}$ always includes the median, say $[\frac{n}{2}]$, of $1, 2, \ldots, n - 1$.

(c) Using the boundary condition (2.10) we obtain

$$(2.11) \qquad h(n) = n(1 + [\log n]) - 2^{[\log n]}.$$

Setting $h(n) = n\Delta S(n)$ we find that $\Delta S(n) = H(n)$ given in (1.15).

Hence using (2.7) and (2.4) it follows that

$$(2.12) \qquad S(n) = \sum_{j=2}^{n} H(j) = A(n).$$

We summarize the results of this section by the

<u>THEOREM 3</u>. To rank n items any inductive procedure is E-optimal in $\mathcal{J}$ if and only if it compares the n^{th} item with any item whose rank, among $n - 1$ items already ranked, belongs to $\Delta_n^{(E)}$. The common minimum expected number of comparisons for all these E-optimal procedures is given by (2.12).

Now we want to use the above technique to find all M-optimal procedures in $\mathcal{J}$. Let R_I' denote any M-optimal inductive procedure in $\mathcal{J}$ and let

$$(2.13) \qquad \Delta W(n) = M(n|R_I') - M(n-1|R_I')$$

where $W(1|R_I') = 0$. Then $\Delta W(n)$ satisfies the recursive relation for $n \geq 2$

$$(2.14) \qquad \Delta W(n) = 1 + \min_{1 \leq y \leq n-1} (\mathrm{Max}(\Delta W(y), \Delta W(n-y)))$$

and boundary conditions

$$(2.15) \qquad W(0) = W(1) = 0.$$

Thus the problem of finding the M-optimal inductive procedure R_I' is reduced to solving the recursive formula (2.14) and boundary conditions (2.15).

The formula (2.14) is investigated above in Section (1.3) and is shown there that

Letting $h(n) = M(n)$ we find that $M(n) = H(n)$ given in (1.1).

Hence using (2.7) and (2.4) it follows that

$$(2.12) \qquad M(n) = \sum_{j=2}^{n} M(j) = A(n).$$

We summarize the results of this section by the

THEOREM 2. To rank n items any inductive procedure is M-optimal in $\mathcal{D}$ if and only if it compares the n^{th} item with any item whose rank, among $n - 1$ items already ranked, belongs to $\mathcal{D}_n^{(*)}$. The common minimum expected number of comparisons for all these M-optimal procedures is given by (2.12).

Now we want to use the above technique to find all M-optimal procedures in $\mathcal{D}$. Let R_1^M denote any M-optimal inductiveprocedure in $\mathcal{D}$ and let

$$(2.13) \qquad M(n) = M(n|R_1^M) - M(n-1|R_1^M)$$

where $M(1|R_1^M) = 0$. Then $M(n)$ satisfies the recursive relation for $n \geq 2$

$$(2.14) \qquad M(n) = 1 + \min_{1 \leq y \leq n-1} \left(\max(M(y), M(n-y))\right)$$

and boundary conditions

$$(2.15) \qquad M(0) = M(1) = 0.$$

Thus the problem of finding the M-optimal inductive procedure R_1^M is reduced to solving the recursive formula (2.14) and boundary conditions (2.15).

The formula (2.14) is investigated above in Section (1.5) and is shown there that (a') For any $n \geq 2$ an integer $y = y(n)$ will yield the minimum in (2.14) if and only if $y \in \Delta_n^{(M)} = [n - 2^{\{\log n\}}, 2^{\{\log n\}}]$.

(b') $\Delta_n^{(M)}$ includes the median, say $[\frac{n}{2}]$, of $1, 2, \ldots, n - 1$.

(c') Furthermore

$$\Delta W(n) = \{\log n\} \tag{2.16}$$

and hence

$$M(n|R_I') = \sum_{j=2}^{n} \{\log j\} = \sum_{j=2}^{n} \{H(j)\} = M(n) \tag{2.17}$$

where $M(n)$ is given by (2.2).

(d') By Lemma 4 in Section (1.4) we have $\Delta_n^{(E)} \subset \Delta_n^{(M)}$. It follows from (d') that any procedure that is E-optimal in $\mathcal{I}$ is also M-optimal in $\mathcal{I}$.

We can summarize the results after (2.13) by the

<u>THEOREM 4</u>. Any inductive procedure for ranking n items is M-optimal in $\mathcal{I}$ if and only if it compares the n^{th} item with any item whose rank among $n - 1$ items already ranked, belongs to $\Delta_n^{(M)}$. The common maximum number of comparisons for all these M-optimal procedures is given by (2.17).

From (b), (b'), (d') and the definition of R_S we have

<u>Corollary 1</u>. The Steinhaus procedure R_S is both E-optimal and M-optimal in $\mathcal{I}$.

Remark 1. In the Iverson book [7], the R_S procedure above is called 'ranking by insertion'. The author says (and we quote from p. 236 of [7])"For a random distribution ranking by insertion requires fewer comparisons than any other method...." Apart from ambiguity

fewer comparisons than any other method...." Apart from a slightly

p. 220 of [7]) "For a random distribution ranking by insertion requires

called 'ranking by insertion'. The author says (and we quote from

Remark 1. In the Iverson book [7], the R_1 procedure above is

Corollary 1. The Steinhaus procedure R_1 is both R-optimal and M-optimal in $\mathcal{S}$.

From (b), (c'), (d') and the definition of R_1 we have

THEOREM 1. Any inductive procedure for ranking n items is M-optimal in $\mathcal{S}$ if and only if it places the n^{th} item with any item whose rank among $n - 1$ items already ranked, belongs to $S_n^{(M)}$. The common minimum number of comparisons for all these M-optimal procedures is given by (2.17).

We can summarize the results after (2.13) by the

(f') By Lemma 4 in Section (1.4) we have $S_n^{(M)} = S_n^{(R)}$. It follows from (d') that any procedure that is R-optimal in $\mathcal{S}$ is also M-optimal in $\mathcal{S}$.

where $M(n)$ is given by (2.2).

$$M(n;R_1^I) = \sum_{j=2}^{n} \{\log j\} = \sum_{j=2}^{n} M(j) = M(n) \tag{2.17}$$

and hence

$$M(n) = \{\log n\} \tag{2.16}$$

(g') Furthermore

(h') $S_n^{(M)}$ includes the median, say $[\frac{n}{2}]$, of $1, 2, \ldots, n - 1$.

(i') For any $n \geq 2$ an integer $y = y(n)$ will yield the minimum in (2.14) if and only if y is in $S_n^{(M)} = [n - 2^{\{\log n\}}, 2^{\{\log n\}}]$.

of 'fewer comparisons' the above statement is inaccurate. By Corollary 1 the procedure R_S has 'optimal' properties only in the restricted class $\mathcal{J}$. In fact for $n = 5$, the procedure R_{FJ} [5] or the counter-example in [18] has maximum number of comparisons equal to $7 < M(5|R_S) = 8$ and also it has expected number of comparisons equal to $6 + \frac{14}{15} < A(n|R_S) = 7 + \frac{1}{15}$. The paper [16] gives several procedures each of which has smaller expectation and smaller maximum than R_S.

Remark 2. It follows from the definition (1.3) that the E-excess of R_S, denoted by $C(n) \equiv C_{n-1}(n|R_S)$ for the problem of ranking n items, i.e., for $t = n - 1$, is given by

$$(2.18) \qquad C(n) = A(n) - L(n).$$

(For $L(n) = H(n!)$ this coincides with the E-noise of R_S.) From the structure of (2.6) and the solution in (2.12) it follows that $A(n) = L(n|\mathcal{J})$ where $L(n|\mathcal{J}) \equiv L(n-1, n|\mathcal{J})$ is given by (1.48) with $t = n - 1$ and $\mathcal{E}_{n-1} = \mathcal{J}$. Hence (2.18) takes the form

$$(2.19) \qquad C(n) = L(n|\mathcal{J}) - L(n).$$

Similarly for the definition (1.14a), for $t = n - 1$ with $C_{Max}(n) \equiv D_{n-1}(n|R_S)$, the structure of (2.14) and the solution (2.17) it follows that

$$(2.20) \qquad C_{Max}(n) = L_{Max}(n|\mathcal{J}) - L_{Max}(n),$$

where $L_{Max}(n|\mathcal{J}) = L_{Max}(n-1, n|\mathcal{J})$ is given by (1.49) with $t = n - 1$ and $\mathcal{E}_{n-1} = \mathcal{J}$. We can interpret (2.19) and also (2.20) to indicate of 'fair comparisons' the above statement is inaccurate. By Corollary 1 the procedure R_n has 'optimal' properties only in the restricted class $\mathcal{C}$. In fact for $n = 5$, as we mentioned in (2.7), the counter-example in [13] has maximum number of comparisons equal to $7 < M(5, R_5) = 8$ and also it has expected number of comparisons equal to $6 + \frac{14}{15} < A(5, R_5) = 7 + \frac{1}{15}$. The paper [16] gives several procedures each of which has smaller expectation and smaller maximum than R_n.

Remark 2. It follows from the definition (1.) that the 1-excess of R_n, denoted by $C(n) = C_{n-1}(n|R_n)$ for the problem of ranking n items, i.e., for $t = n - 1$, is given by

$$C(n) = A(n) - L(n). \tag{2.18}$$

(For $L(n) = H(n)$ which coincides with the H-value of R_n.) From the structure of (2.6) and the solution in (1.12) it follows that $A(n) = L(n|\mathcal{C})$ where $L(n|\mathcal{C}) = L(n-1, n|\mathcal{C})$ is given by (1.42) with $t = n - 1$ and $\mathcal{C}_{n-1} = \mathcal{C}$. Hence (2.18) takes the form

$$C(n) = L(n|\mathcal{C}) - L(n). \tag{2.19}$$

Similarly for the definition (1.44), i.e. $t = n - 1$ with $C_{Max}(n) = C_{n-1}(n|R_n)$, the structure of (2.18) and the solution (2.17) it follows that

$$C_{Max}(n) = L_{Max}(n|\mathcal{C}) - L_{Max}(n), \tag{2.20}$$

where $L_{Max}(n|\mathcal{C}) = L_{Max}(n-1, n|\mathcal{C})$ is given by (1.49) with $t = n - 1$ and $\mathcal{C}_{n-1} = \mathcal{C}$. We can interpret (2.19) and also (2.20) to indicate that the E-excess (M-excess) of the procedure R_S is caused only by the fact that the class $\mathcal{J}$ is not E-complete (M-complete).

2.3. An explicit expression for $S(n)$ and its asymptotic behavior.

From (2.12) and (1.5) we have

$$(2.21)\qquad S(n) = \sum_{j=2}^{n} H(j) = \sum_{j=2}^{n} \{\log j\} - \sum_{j=2}^{n} \left(\frac{2^{\{\log j\}} - j}{j}\right).$$

Since the first summation on the far RHS is in fact $M(n)$ derived by Steinhaus and given in (2.2) above, then

$$(2.22)\qquad S(n) = (1 + \{\log n\})n - 2^{\{\log n\}} - \sum_{j=2}^{n} \frac{2^{\{\log j\}}}{j}.$$

To find an asymptotic expression for $S(n)$ we write the summation in the RHS of (2.22) in a different form. For $2^{s_j - 1} < j \le 2^{s_j}$ we have $s_j = \{\log j\}$. Let $s \equiv s_n = \{\log n\}$ so that $2^{s-1} < n \le 2^s$. Then

$$\sum_{j=2}^{n} \frac{2^{s_j}}{j} = \sum_{j=2}^{2^{s-1}} \frac{2^{s_j}}{j} + \sum_{j=2^{s-1}+1}^{n} \frac{2^{s_j}}{j}$$

$$= \sum_{i=1}^{s-1} \sum_{j=2^{i-1}+1}^{2^i} \frac{2^{s_j}}{j} + 2^s \sum_{j=2^{s-1}+1}^{n} \frac{2^{s_j}}{j}$$

$$= \sum_{i=1}^{s-1} 2^i \sum_{j=2^{i-1}+1}^{2^i} \frac{1}{j} + 2^s \sum_{j=2^{s-1}+1}^{n} \frac{1}{j}.$$

Hence from (2.2) we obtain

$$(2.23)\qquad S(n) = (1+2^s)n - 2^s - \sum_{i=1}^{s-1} 2^i \sum_{j=2^{i-1}+1}^{2^i} \frac{1}{j} - 2^s \sum_{j=2^{s-1}+1}^{n} \frac{1}{j}.$$

that the E-excess (M-excess) of the procedure R_s is caused only by the fact that the class $\mathcal{C}$ is not E-sufficient (M-sufficient).

2.3. An explicit expression for $\bar{E}(n)$ and its asymptotic behavior.

From (2.12) and (1.3) we have

$$\bar{E}(n) = \sum_{j=2}^{n} E(j) = \sum_{j=2}^{n} \{\log j\} - \sum_{j=2}^{n} \left(\frac{2^{\{\log j\}} - j}{j}\right). \tag{2.21}$$

Since the first summation on the RHS is in fact $M(n)$ derived by Steinhaus and given in (2.2) above, then

$$\bar{E}(n) = (1 + \{\log n\})n - 2^{\{\log n\}} - \sum_{j=2}^{n} \frac{2^{\{\log j\}}}{j}. \tag{2.22}$$

To find an asymptotic expression for $\bar{E}(n)$ we write the summation in the RHS of (2.22) in a different form. For $2^{s_j-1} < j \le 2^{s_j}$ we have $s_j = \{\log j\}$. Let $s = s_n = \{\log n\}$ so that $2^{s-1} < n \le 2^{s}$. Then

$$\sum_{j=2}^{n} \frac{2^{s_j}}{j} = \sum_{j=2}^{2^{s-1}} \frac{2^{s_j}}{j} + \sum_{j=2^{s-1}+1}^{n} \frac{2^{s_j}}{j}$$

$$= \sum_{k=1}^{s-1} \sum_{j=2^{k-1}+1}^{2^k} \frac{2^{s_j}}{j} + 2^{s} \sum_{j=2^{s-1}+1}^{n} \frac{1}{j}$$

$$= \sum_{k=1}^{s-1} 2^{k} \sum_{j=2^{k-1}+1}^{2^k} \frac{1}{j} + 2^{s} \sum_{j=2^{s-1}+1}^{n} \frac{1}{j}.$$

Hence from (2.2) we obtain

$$\bar{E}(n) = (1+s)n - 2^{s} - \sum_{k=1}^{s-1} 2^{k} \sum_{j=2^{k-1}+1}^{2^k} \frac{1}{j} - 2^{s} \sum_{j=2^{s-1}+1}^{n} \frac{1}{j}. \tag{2.23}$$

Now we can use the following result given in [3-page 125]

$$(2.24)\qquad \sum_{j=1}^{m} \frac{1}{j} = \ln m + \gamma - \int_{m}^{\infty} f(x)dx$$

where $\gamma = .577$ is Euler's constant,

$$(2.25)\qquad 0 < \int_{m}^{\infty} f(x)dx < \frac{1}{8m^2} ,$$

and for $m = 1$

$$(2.26)\qquad \int_{1}^{\infty} f(x)dx = \gamma - \frac{1}{2} \doteq .0772.$$

From (2.24) we obtain

$$(2.27)\qquad \sum_{j=m_1+1}^{m_2} \frac{1}{j} = \ln \frac{m_2}{m_1} + \frac{1}{2}(\frac{1}{m_2} - \frac{1}{m_1}) + g(m_2) - g(m_1)$$

where

$$(2.28)\qquad g(y) = \int_{1}^{y} f(x)dx.$$

Setting $m_1 = 2^{i-1}$ and $m_2 = 2^i$ in (2.27) we obtain for the double summation in (2.23)

$$(2.29)\qquad \sum_{i=1}^{s-1} 2^i \sum_{j=2^{i-1}+1}^{2^i} \frac{1}{j} = (2^s-2)\ln 2 - \frac{1}{2}(s-1) + \sum_{i=1}^{s-1} 2^i(g(2^i) - g(2^{i-1})).$$

We note that

$$\sum_{i=1}^{s-1} 2^i(g(2^i) - g(2^{i-1})) = \sum_{i=1}^{s-1} (2^i g(2^i) - 2^{i-1}g(2^{i-1}))$$

$$- \sum_{i=1}^{s-1} 2^{i-1}g(2^{i-1}) = 2^{s-1}g(2^{s-1}) - \sum_{i=1}^{s-1} 2^{i-1}g(2^{i-1}).$$

Now we can use the following result given in [-page 125]

$$(2.24) \qquad \sum_{i=1}^{m}\frac{1}{i} = \ln m + \gamma - \int_m^\infty f(x)dx$$

where $\gamma = .577$ is Euler's constant,

$$(2.25) \qquad 0 < \int_m^\infty f(x)dx < \frac{1}{2m},$$

and for $m = 1$

$$(2.26) \qquad \int_1^\infty f(x)dx = \gamma - \frac{1}{2} = .0772.$$

From (2.24) we obtain

$$(2.27) \qquad \sum_{i=m_1+1}^{m_2}\frac{1}{i} = \ln\frac{m_2}{m_1} + \frac{1}{2}\left(\frac{1}{m_2} - \frac{1}{m_1}\right) + g(m_2) - g(m_1)$$

where

$$(2.28) \qquad g(t) = \int_1^t f(x)dx.$$

Setting $m_1 = 2^{i-1}$ and $m_2 = 2^i$ in (2.27) we obtain for the double summation in (2.2)

$$(2.29) \qquad \sum_{i=1}^{s-1} 2^i \sum_{j=2^{i-1}+1}^{2^i}\frac{1}{j} = (2^s-2)\log 2 - \frac{1}{2}(s-1) + \sum_{i=1}^{s-1} 2^i\left(g(2^i) - g(2^{i-1})\right).$$

We note that

$$\sum_{i=1}^{s-1} 2^i\left(g(2^i) - g(2^{i-1})\right) = \sum_{i=1}^{s-1}\left(2^i g(2^i) - 2^{i-1} g(2^{i-1})\right)$$

$$- \sum_{i=1}^{s-1} 2^{i-1} g(2^{i-1}) = 2^{s-1} g(2^{s-1}) - \sum_{i=1}^{s-1} 2^{i-1} g(2^{i-1}).$$

Also letting $m_1 = 2^{s-1}$ and $m_2 = n$ in (2.27) we obtain

$$(2.30) \qquad \sum_{j=2^{s-1}+1}^{n} \frac{1}{j} = \ln 2 - (s-1)\ln 2 + \frac{1}{2}\left(\frac{1}{n} - \frac{1}{2^{s-1}}\right) + g(n) - g(2^{s-1}).$$

Substituting (2.29) and (2.30) in (2.23) we get

$$(2.31) \qquad S(n) = sn - 2^s \ln n + s2^s \ln 2 + n - (1+2 \ln 2)2^s$$
$$+ \frac{1}{2}s - \frac{2^{s-1}}{n} + \frac{1}{2} + 2 \ln 2 + \left(\sum_{i=1}^{s} 2^{i-1}g(2^{i-1}) - 2^s g(n)\right.$$

Using (2.26) and (2.25) we get

$$(2.32) \qquad - g(n) \leq \frac{1}{2} - \gamma + \frac{1}{8n^2}$$

and

$$(2.33) \qquad g(2^{i-1}) \leq \int_{2^{i-1}}^{\infty} f(x)dx + g(2^{i-1}) = \gamma - \frac{1}{2} .$$

Multiplying (2.33) by 2^{i-1} and summing up, and adding the result to 2^r times (2.32) gives

$$- 2^s g(n) + \sum_{i=1}^{s} 2^{i-1}g(2^{i-1}) \leq \frac{1}{2} - \gamma + \frac{2^s}{8n^2} \leq \frac{1}{2} - \gamma + \frac{1}{4n} .$$

Substituting the last result in (2.31) and noting that $\frac{2^{s-1}}{n} < 1$ gives

$$(2.34) \qquad S(n) = sn - 2^s \ln n + s2^s \ln 2 + n - (1+2 \ln 2)2^s + \frac{1}{2} s + \mathcal{O}(1).$$

In particular, for $n = 2^s$ we have $s = \frac{\ln n}{\ln 2} = \log n$ and (2.34) takes the form

$$(2.35) \qquad S(n) = n \log n - (2 \ln 2)n + \frac{1}{2} \log n + \mathcal{O}(1).$$

Expressions similar to the RHS of (2.35) but with fewer terms were obtained by Kislycyn in [8] and [9].

Also letting $n_1 = 2^{s-1}$ and $n_2 = n$ in (2.27) we obtain

$$(2.30)\qquad \sum_{i=2^{s-1}+1}^{n} \frac{1}{i} = \ln n - (s-1)\ln 2 + \frac{1}{2}\left(\frac{1}{n} - \frac{1}{2^{s-1}}\right) + R(n) - R(2^{s-1}).$$

Substituting (2.29) and (2.30) in (2.28) we get

$$(2.31)\qquad s(n) = sn - 2^{s} \ln n + s2^{s} \ln 2 + n - (1+2 \ln 2)2^{s}$$

$$+ \frac{2^{s}}{2} - \frac{2^{s-1}}{n} + \frac{n}{2} + 2 \ln 2 + \left(\sum_{i=1}^{s} 2^{i-1}R(2^{i-1}) - 2^{s}R(n)\right).$$

Using (2.20) and (2.23) we get

$$(2.32)\qquad -R(n) \leq \frac{1}{2} - \gamma + \frac{1}{8n^2}$$

and

$$(2.33)\qquad R(2^{i-1}) \leq \int_{2^{i-1}}^{\infty} f(x)dx + R(2^{i-1}) = \gamma - \frac{1}{2}.$$

Multiplying (2.33) by 2^{i-1} and summing on i, and adding the result to 2^{s} times (2.32) gives

$$-2^{s}R(n) + \sum_{i=1}^{s} 2^{i-1}R(2^{i-1}) \leq \frac{1}{2} - \gamma + \frac{2^{s}}{8n^2} \leq \frac{1}{2} - \gamma + \frac{1}{4n}.$$

Substituting the last result in (2.31) and noting that $\left|\frac{2^{s-1}}{n}\right| \leq 1$ gives

$$(2.34)\qquad s(n) = sn - 2^{s} \ln n + s2^{s} \ln 2 + n - (1+2 \ln 2)2^{s} + \frac{1}{2} s + O(1).$$

In particular, for $n = 2^{s}$ we have $s = \frac{\ln n}{\ln 2} = \log_2 n$ and (2.34) takes the form

$$(2.35)\qquad s(n) = n \log_2 n - (2 \ln 2)n + \frac{1}{2} \log_2 n + O(1).$$

Expressions similar to the RHS of (2.35) but with fewer parameters obtained by Kislyakov in [ ] and [3].

Remark 1. We could also define a procedure, for ranking n items, as being asymptotically E-efficient (M-efficient) if its E-excess (M-excess) approaches zero as $n \to \infty$. By this definition we want to show that R_S is not asymptotically E-efficient (M-efficient).

To establish this claim we make use of $M(n|R_{FJ}) \equiv U(n)$ given in (3.28) below. Of course $U(n)$ being the maximum number of comparisons under a given procedure we have

$$(2.36) \qquad L(n) \leq U(n).$$

Hence

$$(2.37) \qquad C(n) = L(n|\mathcal{J}) - L(n) \geq S(n) - U(n).$$

To show that $C(n)$ does not tend to zero as $n \to \infty$, we take the subsequence $n_k = \frac{1}{3}(2^{2k+2}-1)$ and prove that $S(n_k) - U(n_k)$ approaches $+\infty$ as $k \to \infty$. Using $U(n_k) = n_k(2k-1) + k + 1$ as in (3.33) and substituting $s = \{\log n_k\} = 2k + 1$ in (2.35) we obtain

$$(2.38) \qquad S(n_k) - U(n_k) = 2^{2k+1}(2k+2-2 \ln 2 - \log n_k)\ln 2 + k + \mathcal{O}(1).$$

To finish the proof it suffices to show that

$$(2.39) \qquad 2k + 2 - 2 \ln 2 - \log n_k > 0.$$

Since $n_k < \frac{2^{2k+2}}{3}$ then $\log n_k < 2k + 2 - \log 3$, and since $1.4 \doteq 2 \ln 2 < \log 3 \doteq 1.5$ then (2.39) is true, which proves that R_S is not asymptotically E-efficient.

To prove that R_S is not asymptotically E-efficient we note that

$$(2.40) \qquad C_{Max}(n) = L_{Max}(n|\mathcal{J}) - L_{Max}(n) \geq M(n) - U(n) \geq S(n) - U(n).$$

Since it was shown that $S(n) - U(n)$ does not approach 0 as $n \to \infty$ then it follows from (2.40) that $C_{Max}(n) \not\to 0$ as $n \to \infty$.

Remark 2. Because of the simplicity of R_S, it is frequently used to rank n items with computers. Thus it may be of interest to have an upper bound for its E-excess and M-excess. Using the inequalities (1.9) and (1.15) we obtain

$$(2.41) \qquad C(n) \leq \sum_{j=2}^{n} H(j) - H(n!),$$

and

$$(2.42) \qquad C_{Max}(n) \leq \sum_{j=2}^{n} \{H(j)\} - \{H(n!)\}.$$

To find asymptotic expressions for RHS's of (2.41) and (2.42) we note that by Stirling formula

$$(2.43) \qquad \{H(n!)\} = \{\log n!\} = n \log n - \frac{1}{\ln 2} n + \mathcal{O}(\log n).$$

Using (1.5) we get

$$(2.44) \qquad H(n!) \geq \{\log n!\} - 1 = n \log n - \frac{1}{\ln 2} n + \mathcal{O}(\log n).$$

Hence using (2.34), (2.35), (2.44), (2.43) and (2.17) we obtain

$$(2.45) \qquad C(n) \leq \begin{cases} (\frac{1}{\ln 2} - 2 \ln 2)n + \mathcal{O}(\log n) & \text{for } n = 2^s \\ (\frac{1}{\ln 2} + 1 - \ln 2)n + \mathcal{O}(\log n) & \text{for } n \neq 2^s \end{cases}$$

and

$$(2.46) \qquad C_{Max}(n) \leq \begin{cases} (\frac{1}{\ln 2} - 1)n + \mathcal{O}(\log n) & \text{for } n = 2^s \\ \frac{n}{\ln 2} + \mathcal{O}(\log n) & \text{for } n \neq 2^s. \end{cases}$$

Since it was shown that $\bar{U}(n) - \bar{U}(k)$ does not approach 0 as $n \to \infty$ then it follows from (2.40) that $CM_{max}(n) \to 0$ as $n \to \infty$.

Remark 2. Because of the simplicity of R_n, it is frequently used to rank n items with computers. Thus it may be of interest to have an upper bound for its R-excess and M-excess. Using the inequalities (1.9) and (1.11) we obtain

$$(2.41) \qquad C(n) \le \sum_{j=2}^{n} H(j) - H(n!),$$

and

$$(2.42) \qquad C_{max}^{M}(n) \le \sum_{j=2}^{n} \{H(j)\} - \{H(n!)\}.$$

To find asymptotic expressions for UB's of (2.41) and (2.42) we note that by Stirling formula

$$(2.43) \qquad \{H(n!)\} = \lceil \log n! \rceil = n \log n - \frac{n}{\ln 2} + O(\log n).$$

Using (1.5) we get

$$(2.44) \qquad H(n!) = \lfloor \log n! \rfloor + 1 = n \log n - \frac{n}{\ln 2} + O(\log n).$$

Hence using (2.34), (2.35), (2.43), (2.44) and (1.11) we obtain

$$(2.45) \qquad C(n) \le \begin{cases} (\frac{1}{\ln 2} - 2 \ln 2)n + O(\log n) & \text{for } n = 2^s \\ (\frac{1}{\ln 2} + \text{[illegible]} \ln 2)n + O(\log n) & \text{for } n \ne 2^s \end{cases}$$

and

$$(2.46) \qquad C_{max}(n) \le \begin{cases} (1 - \frac{1}{\ln 2})n + O(\log n) & \text{for } n = 2^s \\ \frac{n}{\ln 2} + O(\log n) & \text{for } n \ne 2^s. \end{cases}$$

2.4. On an inductive procedure $R_S^{(t)}$.

As a first step toward finding a procedure for the problem stated in Section 1.1 with general t we restrict ourselves to a subclass $\mathcal{I}'_t$ of the class of inductive procedures $\mathcal{I}_t$, defined in section (1.5). $\mathcal{I}'_t$ is the class of procedures that rank the t largest of $n - 1$ items inductively and then insert the n^{th} item among the t largest that are already ranked. In $\mathcal{I}_t$ the n^{th} item can be compared with any of $n - t - 1$ other items that are not among the t largest, but not in $\mathcal{I}'_t$.

Let $R_S^{(t)}$ be an E-optimal procedure in $\mathcal{I}'_t$ and let $S_t(n) \equiv A_t(n|R_S^{(t)})$ for fixed t with $1 \le t \le n - 1$. The procedure $R_S^{(t)}$ is defined by the recursive relation for $n \ge t + 1 \ge 2$

$$S_t(n) - S_t(n-1) = 1 + \min_{1 \le y \le t} \left(\frac{y}{n} \cdot H(y) + \frac{n-y}{n}(S_{t-y}(n-y) - S_{t-y}(n-y-1))\right). \tag{2.47}$$

The boundary conditions are

$$S_0(m) = 0 \quad \text{for all } m \tag{2.48}$$

$$S_t(m) = S(m) \equiv A(m|R_S) \qquad t \ge m.$$

In writing $H(x)$ in (2.47) we automatically assumed that the Steinhaus procedure is used when the n^{th} item is larger than the one of rank y among the t largest items already ranked.

Special Cases. For $t = 1$, (2.47) and (2.48) reduce to

$$S_1(n) - S_1(n-1) = 1 \quad \text{with} \quad S_1(1) = 0. \tag{2.50}$$

From (2.50) we get

$$S_1(n) = n - 1 = L(1, n). \tag{2.51}$$

5.4. On an inductive procedure $R_t^{(B)}$.

As a first step toward finding a procedure for the problem stated in Section 1.1 with general t we restrict ourselves to a subclass C_t^n of the class of inductive procedures defined in section (1.5). C_t^n is the class of procedures that rank the t largest of $n-1$ items inductively and then insert the n^{th} item among the t largest that are already ranked. In C_t^n the n^{th} item can be compared with any of $n - t - 1$ other items that are not among the t largest, but not in C_t^n.

Let $R_t^{(B)}$ be an R-optimal procedure in C_t^n and let $g_t(n) = A(n|R_t^{(B)})$ for first t with $1 \le t \le n - 1$. The procedure $R_t^{(B)}$ is defined by the recursive relation for $n \ge t + 1 \ge 2$

$$(5.47) \qquad g_t(n) - g_t(n-1) = 1 + \min_{1 \le y \le t} \left(\frac{y}{t} \cdot H(y) + \frac{t-y}{t}\left(g_{t-1}(t-y) - g_{t-1}(n-y-1)\right)\right).$$

The boundary conditions are

$$(5.48) \qquad g_0(n) = 0 \text{ for all } n$$

$$g_t(n) = g(n) = A(n|R_t) \qquad t \ge n.$$

In writing $H(y)$ in (5.47) we tacitly assumed that the Steinhaus procedure is used when the n^{th} item is larger than the one of rank y among the t largest items already ranked.

<u>Special Case</u>. For $t = 1$, (5.47) and (5.48) reduce to

$$(5.50) \qquad g_1(n) - g_1(n-1) = 1 \text{ with } g_1(1) = 0.$$

From (5.50) we get

$$(5.51) \qquad g_1(n) = n - 1 = L(1, n).$$

For $t = 2$, (2.47) and (2.48) reduce to

$$S_2(n) - S_2(n-1) = 1 + \mathrm{Min}(\frac{n-1}{n}(S_1(n-1) - S_1(n-2)), \frac{2}{n}),$$

or in view of (2.50) we get for $n \geq 3$

$$S_2(n) - S_2(n-1) = 1 + \mathrm{Min}(\frac{n-1}{n}, \frac{2}{n}) = 1 + \frac{2}{n} \tag{2.51}$$

and

$$S_2(2) = 1. \tag{2.52}$$

Since $\frac{2}{n} \leq \frac{n-1}{n}$ for $n \geq 3$ then under the procedure $R_S^{(2)}$ we compare the n^{th} item with the second largest among $n - 1$ items. Expected number of comparisons under $R_S^{(2)}$ is found easily, from (2.51) and (2.52), to be for all $n \geq 2$

$$S_2(n) = n - 1 + 2 \sum_{j=3}^{n} \frac{1}{j} . \tag{2.53}$$

The procedure $R_S^{(2)}$ is the same as R_P in [16-pages 8 and 11].

For $t = n - 1$, (2.47) and (2.48) reduce to (2.6) and (2.7), so that $R_S^{(n-1)} \equiv R_S$.

The study of the recursive relation (2.47), for general t and n, is lengthy enough to be taken up in a separate work.

CHAPTER III

On the Semi-inductive Procedure R_{FJ}

3.1. Introduction.

Ford and Johnson [5] consider the M-optimal problem for ranking n items. Their procedure, which we call R_{FJ}, is in the subclass $\mathcal{J} \equiv \mathcal{J}_{n-1}$, i.e., the class of procedures that do ordinary pairing and use the same procedure on the $[\frac{n}{2}]$ larger items. Let $U(n) = M(n|R_{FJ})$ and $F(n) = A(n|R_{FJ})$. The maximum $U(n)$ is given implicitly in [5] by the recursive relations

$$(3.1) \qquad \begin{cases} U(2k) = k + U(k) + \sum_{i=2}^{k} T(i) \\ U(2k+1) = k + U(k) + \sum_{i=2}^{k+1} T(i) \end{cases}$$

with boundary conditions

$$U(1) = 0, \; U(2) = 1,$$

where

$$(3.2) \qquad T(i) = j \quad \text{for} \quad t_{j-1} < i \leq t_j$$

and

$$(3.3) \qquad t_j = \tfrac{1}{3}(2^{j+1} + (-1)^j) \qquad j = 1, 2, 3, \ldots .$$

It was observed that

$$(3.4) \qquad U(n) = \{H(n!)\} \quad \text{for} \quad n \leq 11 \quad \text{and} \quad n = 20, 21.$$

Hence R_{FJ} is M-optimal for those values of n. It is conjectured in [5] that R_{FJ} is M-optimal for all n. Two asymptotic expressions

CHAPTER III

On the Semi-Inductive Procedure R_{FJ}

1. Introduction.

Ford and Johnson [illegible] consider the M-optimal problem for ranking n items. Their procedure, which we call R_{FJ}, is in the subclass $J = J_{n-1}$, i.e., the class of procedures that do ordinary pairing and use the same procedure on the $[\frac{n}{2}]$ larger items. Let $U(n) = M(n;R_{FJ})$ and $E(n) = A(n;R_{FJ})$. The maximum $U(n)$ is given implicitly in [illegible] by the recursive relations

$$(1.1)\qquad \begin{aligned} U(2k) &= k + U(k) + \sum_{i=2}^{k} T(i) \\ U(2k+1) &= k + U(k) + \sum_{i=2}^{k+1} T(i) \end{aligned}$$

with boundary conditions

$$U(1) = 0,\ U(2) = 1,$$

where

$$(1.2)\qquad T(i) = j \quad \text{for } t_{j-1} < i \leq t_j$$

and

$$(1.3)\qquad t_j = \frac{1}{3}(2^{j+1} + (-1)^j) \qquad j = 1, 2, \ldots\ldots$$

It was observed that

$$(1.4)\qquad U(n) = \{H(n)\} \quad \text{for } n \leq 11 \text{ and } n = 20, 21.$$

Hence R_{FJ} is M-optimal for those values of n. It is conjectured in [illegible] that R_{FJ} is M-optimal for all n. Two asymptotic expressions for $U(n)$ based on subsequences $n_k = [\frac{2^k}{3}]$ and $n_k = 2^k$ where given to make it possible to do some comparisons with $M(n|R_S) = M(n)$.

In this chapter we do the following: 1) prove that $U(n) \leq M(n)$ for all n; 2) we find a single explicit expression for $U(n)$; 3) we find a lower bound $J_{Max}(n|\mathcal{J}) \geq \{H(n!)\}$ for all procedures in the class $\mathcal{J}$; and 4) observe in Table 1 below that $U(n)$ coincides with $L_{Max}(n|\mathcal{J})$ for many other values of n in addition to those in (3.4). We also consider the E-optimal problem in the class $\mathcal{J}$ and for $n = 7$ give a procedure in J that has smaller expectation than R_{FJ}. Thus R_{FJ} is not E-optimal in the class $\mathcal{J}$ and hence it is not E-optimal among all procedures.

The reader should note that in [1-page 229] there is an incorrect description for the R_{FJ}, namely in the second step of the procedure the larger items should be ranked according to the R_{FJ} procedure (semi-inductively) and not by the R_S procedure as stated in [1].

3.2. <u>An explicit expression for $U(n)$.</u>

From (3.1) we have for $k \geq 1$

$$(3.5) \qquad \begin{cases} U(2k) - U(2k-1) = 1 + U(k) - U(k-1) \\ U(2k+1) - U(2k) = T(k+1) \ . \end{cases}$$

Letting

$$(3.6) \qquad V_n = U(n) - U(n-1),$$

then (3.5) can be written as

$$(3.7) \qquad \begin{cases} V_{2k} - V_k = 1 \\ V_{2k+1} = T(k+1) \end{cases}$$

for $U(n)$ based on subsequences $\quad = [\frac{n}{k}]$ and $n_k = 2^k$ where given to make it possible to do some comparisons with $M(n|R_s) = M(n)$.

In this chapter we do the following: 1) prove that $U(n) \leq M(n)$ for all n; 2) we find a single explicit expression for $U(n)$; 3) we find a lower bound $L_{max}(n;J) \leq (M(n!))$ for all procedures in the class J and 4) observe in Table I below that $U(n)$ coincides with $L_{max}(n;J)$ for many other values of n in addition to those in (.4). We also consider the R-optimal procedure in the class J and for $n = 7$ give a procedure in J that has smaller expectation than R_{2^1}. Thus R_{2^1} is not R-optimal in the class J and hence it is not R-optimal among all procedures.

The reader should note that in [1-page 229] there is an incorrect description for the R_{2^1}, namely in the second step of the procedure the larger items should be ranked according to the R_{2^1} procedure (semi-inductively) and not by the R_s procedure as stated in [1].

2. An explicit expression for $U(n)$.

From (.1) we have for $k \geq 1$

$$U(2k) - U(2k-1) = 1 + U(k) - U(k-1) \qquad (.4)$$
$$U(2k+1) - U(2k) = T(k+1) .$$

Letting

$$V_n = U(n) - U(n-1), \qquad (.5)$$

then (.4) can be written as

$$V_{2k} - V_k = 1 \qquad (.6)$$
$$V_{2k+1} = T(k+1)$$

with the boundary condition

$$V_1 = 0. \tag{3.8}$$

Let $\pi_n \equiv \pi(n)$ and $c_n \equiv c(n)$ be non-negative integers defined by (1.47), from which we immediately conclude that

$$\begin{cases} \pi_{2n} = 1 + \pi_n \ , & c_{2n} = c_n \\ \pi_{2n+1} = 0 \ , & c_{2n+1} = n \ . \end{cases} \tag{3.9}$$

Using iteration in (3.7) we get

$$\begin{cases} V_{2k} = 1 + \pi_k + T(c_k+1) \\ V_{2k+1} = T(k+1). \end{cases} \tag{3.10}$$

In view of (3.9) relations (3.10) can be written as

$$\begin{cases} V_{2k} = \pi_{2k} + T(c_{2k}+1) \\ V_{2k+1} = \pi_{2k+1} + T(c_{2k+1}+ 1) \ . \end{cases} \tag{3.11}$$

From (3.11) and (3.8) we conclude that for $n \geq 1$

$$V_n = \pi_n + T(c_n+1) \tag{3.12}$$

where we define

$$T(1) = 0. \tag{3.13}$$

Hence the solution to (3.5) is

$$U(n) = \sum_{i=2}^{n} (\pi_i + T(c_i+1)) \tag{3.14}$$

where $T(k)$ is defined by (3.2) and (3.13).

with the boundary condition

(1.6) $V_1 = 0.$

Let $v_n = v(n)$ and $c_n = c(n)$ be non-negative integers defined by (1.47), from which we immediately conclude that

$$(1.9)\quad \begin{aligned} v_{2n} &= 1 + v_n \;, \quad c_{2n} = c_n \\ v_{2n+1} &= 0 \;, \quad c_{2n+1} = n \;. \end{aligned}$$

Using iteration in (1.7) we get

$$(1.10)\quad \begin{aligned} V_{2k} &= 1 + v_k + T(c_k+1) \\ V_{2k+1} &= T(c_k+1) \;. \end{aligned}$$

In view of (1.9) relations (1.10) can be written as

$$(1.11)\quad \begin{aligned} V_{2k} &= v_{2k} + T(c_{2k}+1) \\ V_{2k+1} &= v_{2k+1} + T(c_{2k+1}+1) \;. \end{aligned}$$

From (1.11) and (1.8) we conclude that for $n \geq 1$

(1.12) $V_n = v_n + T(c_n+1)$

where we define

(1.13) $T(1) = 0.$

Hence the solution to (1.5) is

(1.14) $U(n) = \sum_{j=2}^{n} (v_j + T(c_j+1))$

where $T(k)$ is defined by (1.2) and (1.1).

Now we prove the following lemma which is very useful for comparison of $U(n)$ with $M(n)$. We also use it to find a simpler explicit formula for $U(n)$ in the next section.

Lemma 6. The function T defined by (3.2) and (3.3) and (3.13) is (in explicit form) for $n \geq 1$

$$T(n) = [\log (3n-2)]. \tag{3.15}$$

Proof. Consider n and $j = j(n)$ such that $t_{j-1} < n \leq t_j$ where t_j is the integer given by $t_j = \frac{1}{3}(2^{j+1} + (-1)^j)$. Then

$$2^j + (-1)^{j-1} \leq 3n - 1 < 2^{j+1} + (-1)^j. \tag{3.16}$$

We consider two cases:

(a) $j = 2m$. In this case (3.16) can be written as

$$2^{2m} - 2 \leq 3n - 2 < 2^{2m+1}. \tag{3.17}$$

Since $3m - 2 \equiv -2 \pmod 3$ and $2^{2m} - 2 \equiv -1 \pmod 3$, it follows that we can add 2 to the left side of (3.17) without altering the inequality. Hence

$$2^{2m} \leq 3n - 2 < 2^{2m+1}, \tag{3.18}$$

i.e., $[\log (3n-2)] = 2m = j$ and the lemma is proved in this case.

(b) $j = 2m + 1$. In this case (3.16) can be written as $2^{2m+1} \leq 3n - 1 < 2^{2m+2} - 1$ which implies

$$2^{2m+1} \leq 3n - 2 < 2^{2m+2} - 2 < 2^{2m+2}. \tag{3.19}$$

Hence $[\log (3n-2)] = 2m + 1 = j$ and the lemma is proved.

Now we prove the following lemma which is very useful for comparison of $U(n)$ with $T(n)$. We also use it to find a simple explicit formula for $U(n)$ in the next section.

Lemma [illegible]. The function T defined by (3.2) and (3.3) and (3.1) is (in explicit form) for $n \geq 1$

$$(3.15) \qquad T(n) = [\log (n-2)].$$

Proof. Consider n and $j = j(n)$ such that $t_{j-1} < n \leq t_j$ where t_j is the integer given by $t_j = \frac{1}{3}(2^{j+1} + (-1)^j)$. Then

$$(3.16) \qquad 2^j + (-1)^{j-1} \leq n - 1 < 2^{j+1} + (-1)^j.$$

We consider two cases:

(a) $j = 2m$. In this case (3.16) can be written as

$$(3.17) \qquad 2^{2m} - 2 \leq n - 2 < 2^{2m+1}.$$

Since $n - 2 \equiv -2 \pmod 3$ and $2^{2m} - 2 \equiv -1 \pmod 3$, it follows that we can add 2 to the left side of (3.17) without altering the inequality. Hence

$$(3.18) \qquad 2^{2m} \leq n - 2 < 2^{2m+1},$$

i.e., $[\log (n-2)] = 2m = j$ and the lemma is proved in this case.

(b) $j = 2m + 1$. In this case (3.16) can be written as $2^{2m+1} \leq n - 1 < 2^{2m+2} - 1$ which implies

$$(3.19) \qquad 2^{2m+1} \leq n - 2 < 2^{2m+2} - 2 < 2^{2m+2}.$$

Hence $[\log (n-2)] = 2m + 1 = j$ and the lemma is proved.

Using Lemma 6, we obtain from (3.12)

$$V_n = \pi_n + [\log (3c_n+1)]. \tag{3.20}$$

Now using (1.47) and the fact that $\{\log m\} = 1 + [\log (m-1)]$ we obtain from (3.19)

$$V_n = \pi_n - 1 + [\log (6c_n+2)] = \pi_n - 2 + \{\log 3(2c_n+1)\}$$

$$= - 2 + \{\log 3n\}.$$

Hence

$$V_n = U(n) - U(n-1) = \{\log \tfrac{3n}{4}\}. \tag{3.21}$$

<u>Lemma 7</u>. $U(n) \leq M(n)$ for all n. In particular the inequality is strict if and only if $n \geq 5$.

Proof. From (2.1) and (3.21) we note that $s_n = M(n) - M(n-1) = \{\log n\} \geq \{\log \frac{3n}{4}\} = V_n$. Since $U(n) = \sum_{j=2}^{n} V_j$ and $M(n) = \sum_{j=2}^{n} s_j$ it follows that $U(n) \leq M(n)$ for all n. To finish the rest of the lemma it suffices to show that $n = 5$ is the first integer such that $\{\log \frac{3n}{4}\} < \{\log n\}$. The latter is shown easily by inspection.

3.3. <u>A simpler explicit expression for $U(n)$</u>.

Let for every $i \geq 0$

$$\tau_i = \tfrac{1}{6}(2^{i+2} + (-1)^{i+1} - 3). \tag{3.22}$$

<u>Lemma 8</u>. For V_n in (3.21) and τ_j, $j \geq 1$, in (3.22) have

$$V_n = j - 1 \text{ if } \tau_{j-1} < n \leq \tau_j. \tag{3.23}$$

Using Lemma 3, we obtain from (4.12)

$$(4.20) \qquad V_n = c_n + [\log (c_n+1)].$$

Now using (1.4) and the fact that $[\log m] = 1 + [\log (m-1)]$ we obtain from (4.19)

$$V_n = c_n - 1 + [\log (3c_n+2)] = c_n - 2 + [\log (2c_n+1)]$$

$$= - 2 + [\log n].$$

Hence

$$(4.21) \qquad V_n = U(n) - U(n-1) = [\log \tfrac{n}{4}].$$

Lemma 7. $U(n) \leq M(n)$ for all n. In particular the inequality is strict if and only if $n \geq 5$.

Proof. From (2.1) and (4.21) we note that $a_n = M(n) - M(n-1) = [\log n] \geq [\log \frac{n}{4}] = V_n$. Since $U(n) = \sum_{i=2}^{n} V_i$ and $M(n) = \sum_{i=2}^{n} a_i$ it follows that $U(n) \leq M(n)$ for all n. To finish the rest of the lemma it suffices to show that $n = 5$ is the first integer such that $[\log \frac{n}{4}] < [\log n]$. The latter is shown easily by inspection.

4.4. A simpler explicit expression for $U(n)$.

Let for every $k \geq 0$

$$(4.22) \qquad r_k = \frac{1}{3}(2^{k+2} + (-1)^{k+1} \quad).$$

Lemma 8. For V_n in (4.21) and r_k, $k \geq 1$, in (4.22) have

$$(4.23) \qquad V_n = i - 1 \text{ iff } r_{i-1} < n \leq r_i.$$

Proof. We note that $\tau_j = \frac{1}{2}(t_{j+1}-1)$ where t_j is given by (3.3). Thus if $\tau_{j-1} < n \le \tau_j$ then

$$t_j < 2n + 1 \le t_{j+1}. \tag{3.24}$$

Applying Lemma 6 to (3.24) we get

$$j + 1 = T(2n+1) \tag{3.25}$$

where $T(n)$ is given by (3.15). Hence

$$j + 1 = [\log (6n+1)] = -1 + \{\log (6n+2)\}$$

$$= \{\log (3n+1)\} = [\log 3n] + 1.$$

Hence

$$j = [\log 3n]. \tag{3.25}$$

By the same argument as in Lemma 6 it is easily shown that

$$[\log 3n] - 1 = \{\log \tfrac{3n}{4}\}. \tag{3.26}$$

This completes the proof of the lemma.

Now we proceed to obtain an explicit expression for $U(n)$. Let $\tau_{j-1} < n \le \tau_j$. Then

$$U(n) = \sum_{i=1}^{n} V_i = \sum_{i=1}^{\tau_j} V_i - \sum_{i=n+1}^{\tau_j} V_i \tag{3.27}$$

$$= \sum_{k=1}^{j} \sum_{i=\tau_{k-1}+1}^{\tau_k} V_i - (j-1)(\tau_j - n).$$

For the summation on RHS we have

Proof. We note that $\tau_j = \frac{1}{2}(t_{j+1}-1)$ where t_j is given by (.). Thus if $\tau_{j-1} < n \leq \tau_j$ then

(3.24) $\qquad t_j < 2n + 1 \leq t_{j+1}.$

Applying Lemma 4 to (3.24) we get

(3.2.) $\qquad j + 1 = T(2n+1)$

where $T(n)$ is given by (.1.). Hence

$$j + 1 = [\log (n+1)] = -1 + \{\log (n+2)\}$$

$$= \{\log (n+1)\} = [\log n] + 1.$$

Hence

(3.25) $\qquad j = [\log 2n].$

By the same argument as in Lemma 5 it is easily shown that

(3.26) $\qquad [\log 3n] - 1 = \{\log \frac{n}{..}\}.$

This completes the proof of the lemma.

Now we proceed to obtain an explicit expression for $U(n)$.

Let $\tau_{j-1} < n \leq \tau_j$. Then

(3.27) $\qquad U(n) = \sum_{i=1}^{n} V_i = \sum_{i=1}^{\tau_j} V_i - \sum_{i=n+1}^{\tau_j} V_i$

$$= \sum_{k=1}^{k-1} \sum_{i=\tau+1}^{\tau_k} V_i - (j-1)(\tau_j - n).$$

For the summation on RHS we have

$$\sum_{k=1}^{j} \sum_{i=\tau_{k-1}+1}^{\tau_k} V_i = \sum_{k=1}^{j} (\tau_k - \tau_{k-1})(k-1) = \sum_{k=1}^{j} (k\tau_k - (k-1)\tau_k) - \sum_{k=1}^{j} \tau_k .$$

Hence we obtain for (3.27)

$$U(n) = n(j-1) + \tau_j - \sum_{k=1}^{j} \tau_k = (n+\tfrac{1}{2})(j-1) - \tfrac{1}{2}(\tau_{j-1} - 2 + 2^j).$$

Substituting for τ_{j-1} from (3.22) in the last expression we have for all $n \geq 1$

$$U(n) = n(j-1) + \frac{1}{2} j - \frac{2}{3}(2^j - 1) + \frac{1}{6}\left(\frac{1+(-1)^{j+1}}{2}\right) \tag{3.28}$$

where $j = j(n) = [\log 3n]$.

Although the result for $U(n)$ in [16, eq. (6.15)] is slightly different from $U(n)$ in (3.28) above, it is easy to show that the 2 expressions for $U(n)$ are consistent.

From (3.28) we can readily write an asymptotic expression for $U(n)$, i.e.,

$$U(n) = nj - n\left(1 + \frac{2^{1+j}}{3n}\right) + \frac{1}{2} \log n + \mathcal{O}(1), \tag{3.29}$$

where the coefficient of n in parenthesis lies between 4 and 7, and its particular value depends on the subsequence of n chosen. For the subsequence

$$n_i = \left\{\frac{2^{2i+1}}{3}\right\} = \frac{2^{2i+1}+1}{3} \qquad i = 1, 2, \ldots \tag{3.30}$$

we have $j = 2i + 1$ and by (3.28)

$$U(n) = 2n(i-1) + i + 2. \tag{3.31}$$

For the subsequence

$$\sum_{i=1}^{j-1} \sum_{k=2^i+1}^{2^{i+1}} v_k = \sum_{i=1}^{j-1} (2^i - 2^{i-1})(i-1) = \sum_{i=1}^{j-1} (i2^i - (i-1)2^{i-1}) - \sum_{i=1}^{j-1} 2^i .$$

Hence we obtain for (2.27)

$$U(n) = n(j-1) + 2^j - \sum_{i=1}^{j-1} 2^i = (n + \tfrac{1}{2})(j-1) - \tfrac{2}{3}(2^{j-1} - 2 + 2^j) .$$

Substituting for 2^{j-1} and (2.22) in the last expression we have for all $n \geq 1$

(2.28) $$U(n) = n(j-1) + \frac{j}{2} - \frac{2}{3}(2^{j}-1) + \frac{2}{9}\left(\frac{1+(-1)^{j+1}}{2}\right)$$

where $j = j(n) = \lfloor \log_2 n \rfloor$.

Although the result for $U(n)$ in [Ho, eq. (3.17)] is slightly different from $U(n)$ in (2.28) above, it is easy to show that the 2 expressions for $U(n)$ are consistent.

From (2.28) we can readily write an asymptotic expression for $U(n)$, i.e.,

(2.29) $$U(n) = nj - n\left(1 + \frac{2^{j+1}}{3n}\right) + \frac{1}{2}\log_2 n + O(1).$$

where the coefficient of n in parenthesis lies between 4 and 2, and its particular value depends on the subsequence of n chosen. For the subsequence

(2.30) $$n_i = \frac{2^{2i+1}+1}{3} = \frac{2^{j}+1}{3}, \quad i = 1, 2, \ldots$$

we have $j = 2i + 1$ and by (2.28)

(2.31) $$U(n) = 2n(i-1) + i + 2.$$

For the subsequence

$$(3.32) \qquad n_i = \left[\frac{2^{2i+2}}{3}\right] = \frac{2^{2i+2}-1}{3} \qquad i = 1, 2, \ldots$$

we have $j = 2i + 1$ and by (3.28)

$$(3.33) \qquad U(n) = n(2i-1) + i + 1.$$

3.4. Lower bounds for all procedures in the class $\mathcal{J}$.

First we want to find a lower bound $J_{Max}(n)$ for the maximum number of comparisons needed for ranking n items under any procedure in the class $\mathcal{J}$. For all procedures in J we do the ordinary pairing and then rank the $[\frac{n}{2}]$ winners (or larger items) by the same procedure. Let the $[\frac{n}{2}]$ winners be denoted by $x_2 \leq x_4 \leq x_6 \leq \ldots$. Denote the corresponding losers by $x_1, x_3, x_5, \ldots$ and insert them in that order. Then x_{2i+1} has $2i + 1$ different places (or spaces) in which it can go and hence we are left with $1\cdot 3\cdot 5\cdot \ldots \cdot(2k-1)$ or $1\cdot 3\cdot 5\cdot \ldots \cdot(2k+1)$ states of nature depending on whether $n = 2k$ or $n = 2k + 1$, respectively. We define $J_{Max}(n)$ by the recursive relations

$$(3.34) \qquad \begin{cases} J_{Max}(2k) = k + J_{Max}(k) + \{H(3\cdot 5\cdot \ldots \cdot(2k-1)\} \\ J_{Max}(2k+1) = k + J_{Max}(k) + \{H(3\cdot 5\cdot \ldots \cdot(2k+1))\} \end{cases}$$

and the boundary condition $J_{Max}(1) = 0$.

The values of $J_{Max}(n)$ can now be calculated with the help of (1.5), and they are given in Table 1 below.

Since $\{H(n!)\}$ is a lower bound for any procedure, $J_{Max}(n)$ is a lower bound for procedures in $\mathcal{J}$ and $L_{Max}(n|\mathcal{J})$ is the best that can be achieved, we have for any procedure like R_{FJ} in $\mathcal{J}$

$$(3.35) \qquad \{H(n!)\} \leq J_{Max}(n) \leq L_{Max}(n|\mathcal{J}) \leq U(n).$$

$$(1.2) \qquad n_i = \left[\frac{2^{2i+2}}{3}\right] = \frac{2^{2i+2}-1}{3}, \qquad i = 1, 2, \ldots$$

we have $j = 2i + 1$ and by (1.2)

$$(1.3) \qquad U(n) = n(h-1) + i + 1.$$

<u>3.4. Lower bounds for all procedures in the class $\mathcal{J}$.</u>

First we want to find a lower bound $L_{MK}(n)$ for the minimum number of comparisons needed for ranking n items under any procedure in the class $\mathcal{J}$. For all procedures in $\mathcal{J}$ we do the ordinary pairing and then rank the $[\frac{n}{2}]$ winners (or larger items) by the same procedure. Let the $[\frac{n}{2}]$ winners be denoted by $x_1 \leq x_2 \leq x_3 \leq \ldots$. Denote the corresponding losers by $y_1, y_2, y_3, \ldots$ and insert them in that order. Then x_{2k+1} has $2k + 1$ different places (or spaces) in which it can go and hence we are left with $1 \cdot 3 \cdot \ldots \cdot (2k-1)$ or $1 \cdot 3 \cdot \ldots \cdot (2k+1)$ states of nature depending on whether $n = 2k$ or $n = 2k + 1$, respectively. We define $L_{MK}(n)$ by the recursive relations

$$(1.4) \qquad \begin{aligned} L_{MK}(2k) &= k + L_{MK}(k) + \{lg(1 \cdot 3 \cdot \ldots \cdot (2k-1))\}, \\ L_{MK}(2k+1) &= k + L_{MK}(k) + \{lg(1 \cdot 3 \cdot \ldots \cdot (2k+1))\}, \end{aligned}$$

and the boundary condition $L_{MK}(1) = 0$.

The values of $L_{MK}(n)$ can now be calculated with the help of (1.4), and they are given in Table 1 below.

Since $\{lg(n!)\}$ is a lower bound for any procedure and $L_{MK}(n)$ is a lower bound for procedures in $\mathcal{J}$ and $L_{MK}(n;\mathcal{J})$ is the best that can be achieved, we have for any procedure like R_{MK} in $\mathcal{J}$

$$(1.5) \qquad \{lg(n!)\} \leq L_{MK}(n) \leq L_{MK}(n;\mathcal{J}) \leq U(n).$$

If $U(n) = J_{Max}(n)$ for some n then R_{FJ} is M-optimal in $\mathcal{J}$ for that particular value (or those values) of n. Table 1 shows that R_{FJ} is M-optimal in $\mathcal{J}$ for many values of n including the ones for which R_{FJ} is M-optimal among all procedures. For example R_{FJ} is M-optimal in $\mathcal{J}$ (but not necessarily among all procedures) for $n = 12, 13, 14, 17$ etc.

Table 1

n	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$U(n)$	30	34	38	42	46	50	54	58	62	66	71	76	81	86
$J_{Max}(n)$	30	34	38	41	45	50	54	58	62	66	71	76	81	85
$\{H(n!)\}$	29	33	37	41	45	49	53	57	62	66	70	75	80	84

(For $n \leq 11$, we find that $U(n) = J_{Max}(n) = \{H(n!)\}$.)

At the end of this section an explicit expression for $J_{Max}(n)$ and the subsequence $n_j = 2^j - 1$ is found using recursive relations (3.34) and the boundary condition following (3.34).

Now considering the E-optimal problem for the class $\mathcal{J}$, we find a lower bound $J(n)$ for the expected number of comparisons over all procedures in $\mathcal{J}$. Corresponding to (3.34) we now have

$$(3.36) \qquad \begin{cases} J(2k) = k + J(k) + H(3 \cdot 5 \cdot \ldots \cdot (2k-1)) \\ J(2k+1) = k + J(k) + H(3 \cdot 5 \cdot \ldots \cdot (2k+1)) \end{cases}$$

with the boundary condition $J(1) = 0$.

By a similar argument to that used for (3.35) we have

$$(3.37) \qquad H(n!) \leq J(n) \leq L(n|\mathcal{J}) \leq F(n)$$

If $U(n) = J_{MAX}(n)$ for some n then R_{ST} is M-optimal in J for that particular value (or those values) of n. Table I shows that R_{ST} is M-optimal in J for many values of n including those for which R_{ST} is M-optimal among all procedures. For example R_{ST} is M-optimal in J (but not necessarily among all procedures) for $n = 12, 13, 15, 17$ etc.

Table I

n	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$J_{MAX}(n)$	[illegible]	[illegible]	38	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$M(n)$	[illegible]	[illegible]	[illegible]	41	[illegible]	50	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$H(n!)$	29	33	[illegible]	41	45	[illegible]	[illegible]	[illegible]	62	[illegible]	70	[illegible]	[illegible]	[illegible]

(For $n \leq 11$, we find that $U(n) = J_{MAX}(n) = \lceil H(n!) \rceil$.)

At the end of this section an explicit expression for $J_{MAX}(n)$ and the subsequence $n_i = 2^i - 1$ is found using recursive relations (3.?) and the boundary condition following (3.?).

Now considering the M-optimal problem for the class J, we find a lower bound $J(n)$ for the expected number of comparisons over all procedures in J. Corresponding to (3.?) we have

$$J(2k) = k + J(k) + H(\ldots(2k-1))$$
$$J(2k+1) = k + J(k) + H(\ldots(2k+1)) \qquad (3.15)$$

with the boundary condition $J(1) = 0$.

By a similar argument to that used for (3.?) we have

$$H(n!) \leq J(n) \leq L(n) \leq K(n) \qquad (3.17)$$

where $F(n) \equiv A(n|R_{FJ})$.

In Table 2 below few values of $J(n)$, derived from (3.36) are given. The values of $F(n)$ were calculated using the definition of R_{FJ} in [5] and are the same values that appear in [16].

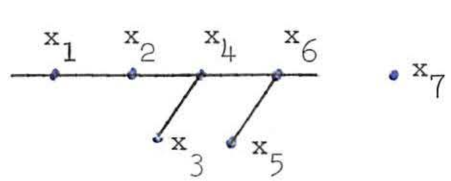
Table 2

n	2	3	4	5	6	7	8	9
$F(n)$	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{27}{45}$	$12\frac{144}{315}$	$15\frac{144}{315}$	$18\frac{1656}{2835}$
$J(n)$	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{27}{45}$	$12\frac{141}{315}$	$15\frac{141}{315}$	$18\frac{1653}{2835}$
$H(n!)$	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{118}{315}$	$15\frac{118}{315}$	$18\frac{1574}{2835}$

(Values of $F(n)$ for $n = 10, 11, \ldots, 16$ are given in Table 4 below.)

If $F(n) = J(n)$ for some value(s) of n then R_{FJ} is E-optimal in the class $\mathcal{J}$ for those particular value(s). In Section (1.2) we pointed out that R_{FJ} is not E-optimal among all procedures since we had a counter-example for $n = 6$. In fact the table in [16-Section 5] shows that there are several other procedures with smaller expectation than $F(n)$ for $n \leq 10$. We now show, in addition, that R_{FJ} is not E-optimal in the class $\mathcal{J}$. The tree of Fig. 4 represents a procedure of $\mathcal{J}$ that is E-optimal in $\mathcal{J}$ for $n = 7$; it has expectation $12 + \frac{141}{315} < F(7) = 12 + \frac{144}{315}$.

For $n = 7$ all the procedures in $\mathcal{J}$ start with ordinary pairing and ranking of the larger items which requires an average of $5 + \frac{2}{3}$ comparisons and yield the following diagram



where $T(n) = A(n|E_{T1})$.

In Table 2 below few values of $J(n)$, derived from (1.6) are given. The values of $T(n)$ were calculated using the definition of E_{T1} in [?] and are the same values that shown in [10].

Table 2

n	2	3	4	5	6	7	8	9
$T(n)$	1	[illegible]	[illegible]	[illegible]	[illegible]	12 [illegible]	[illegible]	13 [illegible]
$J(n)$	1	[illegible]	[illegible]	[illegible]	[illegible]	12 [illegible]	[illegible]	[illegible]
$E(n)$	1	[illegible]	[illegible]	[illegible]	[illegible]	12 [illegible]	[illegible]	[illegible]

(Values of $T(n)$ for $n = 10, 11, \ldots, 15$ are given in Table 4 below.)

If $T(n) = J(n)$ for some value(s) of n then E_{T1} is E-optimal among class $\mathcal{J}_n$ for those particular value(s). In Section (1.2) we pointed out that E_{T1} is not E-optimal among all procedures since we had a counter-example for $n = 6$. In fact the table in [?-Section ?] shows that there are several other procedures with smaller expectation than $T(n)$ for $n \leq 10$. We now show, in addition, that E_{T1} is not E-optimal in the class $\mathcal{J}_n$. The tree of E_{T1} represents a procedure of $\mathcal{J}_n$ that is E-optimal in $\mathcal{J}_n$ for $n = 7$; its expectation

$$12 + \frac{141}{15} < E(7) = 12 + \frac{144}{15}.$$

For $E1 = 7$ all the procedures in $\mathcal{J}_n$ start with ordinary-pairing and ranking of the larger ones which requires an a-rate of $+\frac{?}{?}$ comparisons and yield the following diagram

A continuation which leads to a procedure that is in $\mathcal{G}$ is given below in Fig. 4.

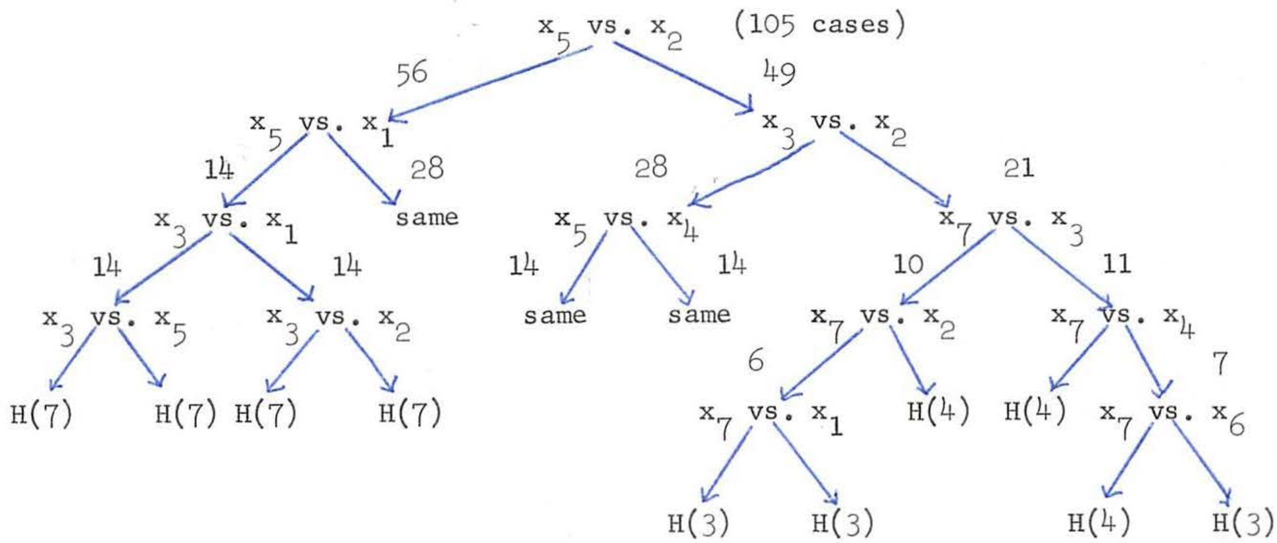


Fig. 4. <u>A continuation that is E-optimal in $\mathcal{G}$</u>.

Now we want to find explicit expressions for $J_{Max}(n)$ and $J(n)$ using recursive relations (3.34) and (3.36) respectively. It will be clear that (3.34) and (3.36) are special cases of the recursive formula (3.39) below.

Define δ_x by

(3.38) $\qquad \delta_x$ = the largest odd integer not smaller than n.

Consider for any function f with $f(1) = 0$ the recursive relation

$$I(n) = [\tfrac{n}{2}] + I([\tfrac{n}{2}]) + f(\delta_n) \tag{3.39}$$

and boundary condition $I(1) = 0$. We note that if $f(\delta_n) = \{H(3\cdot 5\cdot \ldots \cdot \delta_n)\}$ then (3.39) reduces to (3.34), and if $f(\delta_n) = H(3\cdot 5 \cdot \ldots \cdot \delta_n)$ then (3.39) reduces to (3.36). Hence $J_{Max}(n)$ and $J(n)$ can be found from $I(n)$.

A construction which leads to a procedure that is in $\mathcal{T}$ is given below in Fig. 4.

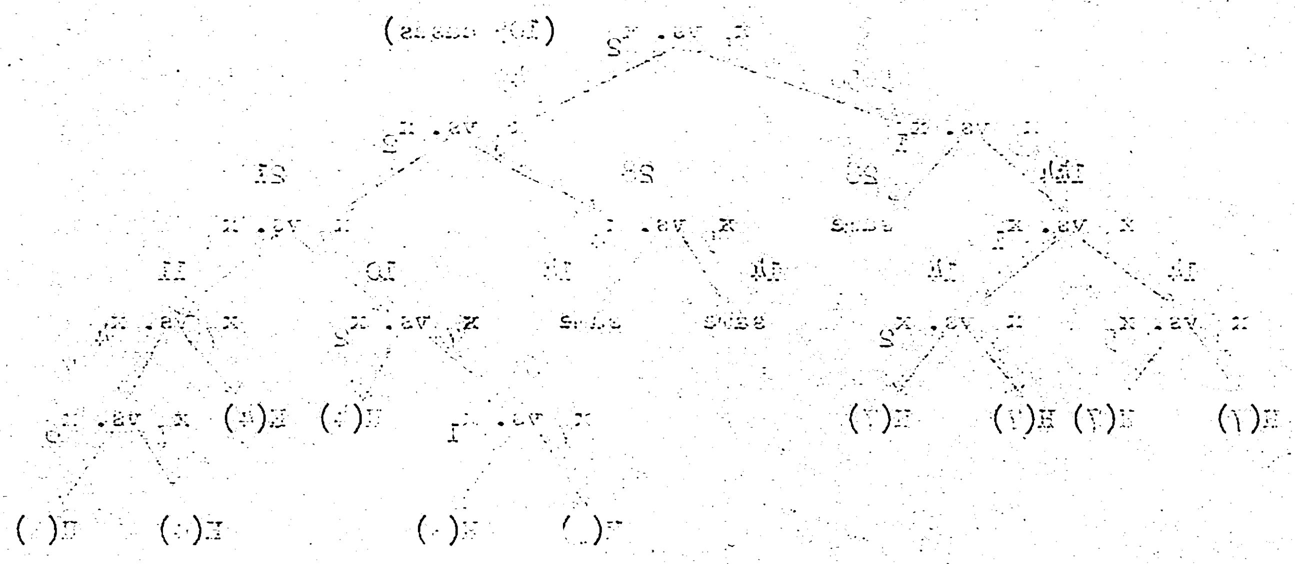


Fig. 4. A construction that is K-optimal in [illegible]

Now we want to find explicit expressions for $T_{max}(n)$ and $T(n)$ using recursive relations ([illegible]) and ([illegible]) respectively. It will be clear that ([illegible]) and ([illegible]) are special cases of the recursive formula ([illegible]) below.

Define $\bar{n}$ by

([illegible]) $\bar{n}$ = the largest odd integer (or smaller than n.

Consider for any function f with $f(1) = 0$ the recursive relation

$$([illegible]) \quad T(n) = [\tfrac{n}{2}] + T([\tfrac{n}{2}]) + f(\bar{n})$$

and boundary condition $T(1) = 0$. We note that if $f(\bar{n}) = H(\cdot\cdot\ldots\cdot\bar{n})$ then ([illegible]) reduces to ([illegible]), and if $f(\bar{n}) = H(\cdot\cdot\ldots\cdot\bar{n})$ then ([illegible]) reduces to ([illegible]). Hence $T_{max}(n)$ and $T(n)$ can be found [illegible].

To solve (3.39) for $I(n)$ we need to show that

$$(3.40)\qquad \underbrace{[\ldots\tfrac{1}{2}[\tfrac{1}{2}[\tfrac{1}{2}}_{\text{j brackets}}\,n]]\ldots] = [\frac{n}{2^j}].$$

We show (3.40) using induction on j. For $j = 1$ (3.40) is trivial. It remains to show that

$$(3.41)\qquad [\tfrac{1}{2}[\frac{n}{2^{j-1}}]] = [\frac{n}{2^j}].$$

Case 1. $n < 2^j$, i.e., $j > [\log n]$ then the RHS of (3.41) is equal to zero. Since $0 \leq \frac{n}{2^{j-1}} < 2$ then $0 \leq [\frac{n}{2^{j-1}}] < 2$ and hence

$$(3.42)\qquad 0 \leq \tfrac{1}{2}\,[\frac{n}{2^{j-1}}] < 1$$

which shows that the LHS of (3.41) is also zero in this case.

Case 2. $n \geq 2^j$, i.e., $j \leq [\log n]$. Let non-negative integers β and ε be defined such that

$$(3.43)\qquad n = \beta 2^j + \varepsilon \qquad 0 \leq \varepsilon < 2^j.$$

Then for the RHS of (3.41) we have $[\frac{n}{2^j}] = [\frac{\beta 2^j+\varepsilon}{2^j}] = \beta$. To find the LHS of (3.42) we note from (3.43) that

$$(3.44)\qquad 2\beta \leq \frac{n}{2^{j-1}} < 2(\beta+1).$$

Hence

$$(3.45)\qquad \beta \leq \tfrac{1}{2}\,[\frac{n}{2^{j-1}}] < \beta + 1$$

which implies $\beta = [\frac{1}{2}[\frac{n}{2^{j-1}}]]$. Thus (3.41) is true.

To solve (4.?) for $T(n)$ we need to show that

$$(4.40) \qquad \underbrace{[\ldots[\tfrac{1}{2}[\tfrac{1}{2}[\tfrac{1}{2}n]]\ldots]}_{j \text{ brackets}} = [\frac{n}{2^j}].$$

We show (4.40) using induction on j. For $j = 1$ (4.40) is trivial. It remains to show that

$$(4.41) \qquad [\frac{1}{2}[\frac{n}{2^{j-1}}]] = [\frac{n}{2^j}].$$

<u>Case 1.</u> $n < 2^j$, i.e., $j > [\lg n]$ then the RHS of (4.41) is equal to zero. Since $0 \le \frac{n}{2^{j-1}} < 2$ then $0 \le [\frac{n}{2^{j-1}}] < 2$ and hence

$$(4.42) \qquad 0 \le \frac{1}{2}[\frac{n}{2^{j-1}}] < 1$$

which shows that the LHS of (4.41) is also zero in this case.

<u>Case 2.</u> $n \ge 2^j$, i.e., $j \le [\lg n]$ (say). Let non-negative integers k and r be defined such that

$$(4.43) \qquad n = k2^j + r \qquad 0 \le r < 2^j.$$

Then for the RHS of (4.41) we have $[\frac{n}{2^j}] = [\frac{k2^j+r}{2^j}] = k$. To find the LHS of (4.42) we note from (4.43) that

$$(4.44) \qquad 2k \le \frac{n}{2^{j-1}} < 2(k+1).$$

Hence

$$(4.45) \qquad k \le \frac{1}{2}[\frac{n}{2^{j-1}}] < k + 1$$

which implies $k = [\frac{1}{2}[\frac{n}{2^{j-1}}]]$. Thus (4.41) is true.

Using (3.40) and iteration in (3.39) we obtain

$$(3.46)\qquad I(n) = \sum_{j=1}^{[\log n]} [\frac{n}{2^j}] + \sum_{j=0}^{[\log n]-1} f(\lambda_{[\frac{n}{2^j}]}).$$

Using (1.46) and the fact that $\lambda_x = \lambda_{[x]}$ we have

$$(3.47)\qquad I(n) = n - \alpha + \sum_{j=0}^{[\log n]-1} f(\lambda_{\frac{n}{2^j}})$$

where α is the number of 1's in the binary expansion of n.

Letting $f(\lambda_{\frac{n}{2^j}}) = \{H(3\cdot 5\cdot \ldots \cdot\lambda_{\frac{n}{2^j}})\}$ in (3.47) we obtain

$$(3.48)\qquad J_{Max}(n) = n - \alpha + \sum_{j=0}^{[\log n]-1} \{H(3\cdot 5\cdot \ldots \cdot\lambda_{\frac{n}{2^j}})\},$$

and letting $f(\lambda_{\frac{n}{2^j}}) = H(3\cdot 5\cdot \ldots \cdot\lambda_{\frac{n}{2^j}})$ in (3.47) we obtain

$$(3.49)\qquad J(n) = n - \alpha + \sum_{j=0}^{[\log n]-1} H(3\cdot 5\cdot \ldots \cdot\lambda_{\frac{n}{2^j}}).$$

In particular for the subsequence $n_k = 2^k$ we have $\lambda_{\frac{n}{2^j}} = 2^{k-j} - 1$, $\alpha = 1$ and hence

$$(3.50)\qquad J_{Max}(n_k) = n_k - 1 + \sum_{j=0}^{k-1} \{H(3\cdot 5\cdot \ldots \cdot(2^{k-j}-1))\}$$

and

$$(3.51)\qquad J(n_k) = n_k - 1 + \sum_{j=0}^{k-1} H(3\cdot 5\cdot \ldots \cdot(2^{k-j}-1)).$$

Finally we note that

$$(3.52)\qquad J_{Max}(n_k - 1) = J_{Max}(n_k) - k - 1$$

Using (1.15) and iteration in (1.5) we obtain

$$(1.46) \qquad l(n) = \sum_{j=[\log n]}^{1} \left[\frac{S_j}{n}\right] + \sum_{j=[\log n]-1}^{0} E\left(y^{\frac{S_j}{n}}\right).$$

Using (1.46) and the fact that $y^{2} = y^{[2]}$ we have

$$(1.47) \qquad l(n) = n - m + \sum_{j=[\log n]-1}^{0} E\left(y^{\frac{S_j}{n}}\right)$$

where m is the number of 1's in the binary expansion of n.

Letting $E\left(y^{\frac{S_j}{n}}\right) = [E(2,\ldots,y^{\frac{S_j}{n}})]$ in (1.47) we obtain

$$(1.48) \qquad l^{bin}(n) = n - m + \sum_{j=[\log n]-1}^{0} E\left(2,\ldots,y^{\frac{S_j}{n}}\right);$$

and letting $E\left(y^{\frac{S_j}{n}}\right) = E\left(2,\ldots,y^{\frac{S_j}{n}}\right)$ in (1.47) we obtain

$$(1.49) \qquad l(n) = n - m + \sum_{j=[\log n]-1}^{0} E\left(2,\ldots,y^{\frac{S_j}{n}}\right).$$

In particular for the subsequence $n^k = 2^k$ we have $y^{\frac{S_j}{n}} = 2^{j-1} - 1, j = 1$

and hence

$$(1.50) \qquad l^{bin}(n^k) = n^k - 1 + \sum_{j=1}^{k} E\left(2,1,\ldots,(2^{k-j}-1)\right);$$

and

$$(1.51) \qquad l(n^k) = n^k - 1 + \sum_{j=1}^{k} E\left(\ldots,(2^{j-1}-1)\right).$$

Finally we note that

$$(1.52) \qquad l^{bin}(n^k - 1) = l^{bin}(n^k) - k - 1$$

and

$$(3.53) \qquad J(n_k-1) = J(n_k) - k - 1.$$

3.5. Recursive relations for $F(n) \equiv A(n|R_{FJ})$.

After ordinary pairing and semi-induction for $[\frac{n}{2}]$ larger items we are left with one of the two configurations (a) or (b) in Fig. 5, depending upon whether $n = 2k$ or $n = 2k + 1$ respectively.

(a)

(b)

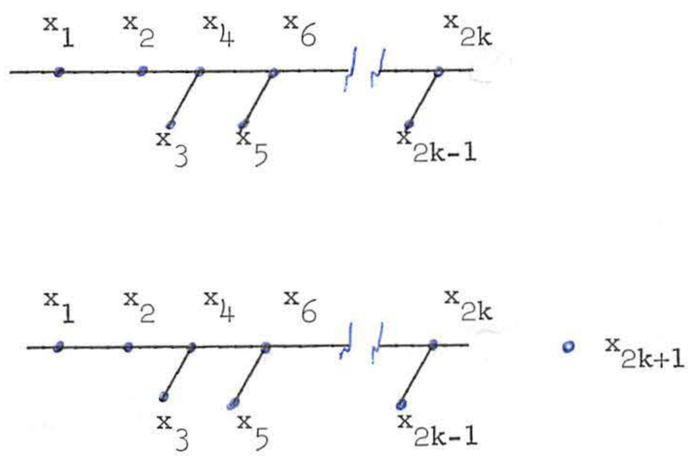


Fig. 5

Let the order of insertion of the items with indices $3, 5, \ldots, 2[\frac{n-1}{2}] + 1$, according to the R_{FJ} procedure given in [5], be $1, 2, 3, \ldots, [\frac{n-1}{2}]$, respectively. For convenience we denote the item to be inserted first by y_1, the next one to be inserted by y_2, etc., and the last to be inserted by $y_{[\frac{n-1}{2}]}$. For example, if $n = 15$, then the items to be inserted are $x_3, x_5, \ldots, x_{15}$ and the order of insertion under R_{FJ} is $x_5, x_3, x_9, x_7, x_{15}, x_{13}, x_{11}$; hence $y_1 = x_5$, $y_2 = x_3, \ldots, y_7 = x_{11}$.

Let $e_n(\alpha)$ be the expected number of comparisons needed for the insertion of y_α given that the items $y_1, y_2, \ldots, y_{\alpha-1}$ are inserted.

and

$$f(n_k-1) = f(n_k) - k - 1. \qquad (4.4)$$

5. <u>Recursive relations for $V(n) \equiv A(n,K_n)$.</u>

After ordinary pairing and semi-induction for $[\frac{n}{2}]$ larger items we are left with one of the two configurations (a) or (b) in Fig. 1, depending upon whether $n = 2k$ or $n = 2k + 1$ respectively.

(a)

(b)

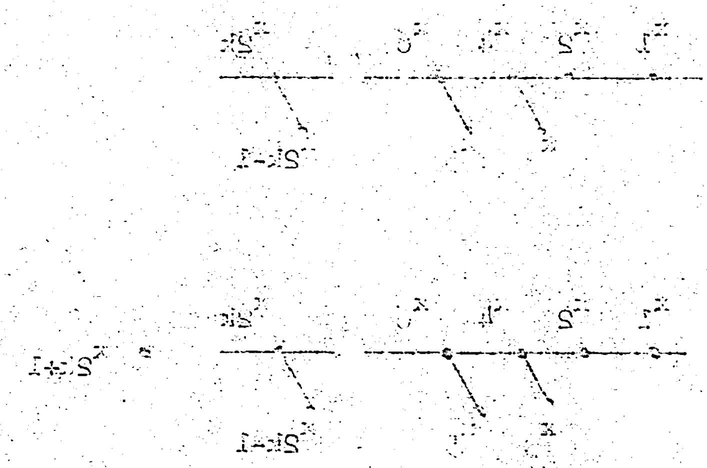


Fig. 1

Let the order of insertion of the items with indices [illegible], ..., $2[\frac{n-1}{2}] + 1$, according to the $F_{[illegible]}$ procedure given in [1], be 1, 2, ..., $[\frac{n-1}{2}]$, respectively. For convenience, we denote the item to be inserted first by y_1, the next one to be inserted by y_2, etc., and the last to be inserted by $y_{[\frac{n-1}{2}]}$. For example, if $n = 14$, then the items to be inserted are $x_3, x_5, \ldots, x_{13}$ and the order of insertion under F_{14} is [illegible]; hence $y_1 = x_5$, $y_2 = x_3, \ldots, y_6 = x_{13}$.

Let $e_n(i)$ be the expected number of comparisons needed for the insertion of y_i given that the items $y_1, y_2, \ldots, y_{i-1}$ are inserted.

Then for $n \geq 2$

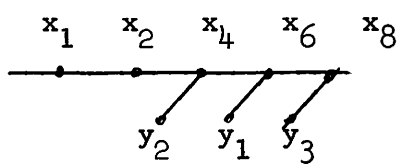
$$(3.54) \qquad F(n) = [\tfrac{n}{2}] + F([\tfrac{n}{2}]) + \sum_{\alpha=1}^{[\frac{n-1}{2}]} e_n(\alpha)$$

and $F(1) = 0$.

The values of $e_n(\alpha)$ for small α are easy to compute but for large n (say $n > 16$) and large α (say $\alpha > 7$) the calculations are tedious. For example, we have for $n = 8$

$$e_8(1) = H(4),\ e_8(2) = \tfrac{1}{5} H(3) + \tfrac{4}{5} H(4) \quad \text{and} \quad e_8(3) = H(7)$$

where $H(n)$ is given by (1.5). To see this we consider the configuration



It is clear that $e_8(1) = H(4)$ which is the expected number of comparisons needed for insertion of y_1 into the chain $x_1 < x_2 < x_4$ under the R_S procedure. If $x_4 < y_1$ (an event with probability $\frac{3 \cdot 7}{3 \cdot 5 \cdot 7} = \frac{1}{5}$) then $H(3)$ is the expected number of comparisons for insertion of y_2 in the chain $x_1 < x_2$. If $y_1 < x_4$ (an event with probability $\frac{3 \cdot 4 \cdot 7}{3 \cdot 5 \cdot 7} = \frac{4}{5}$) then $H(4)$ comparisons is the average needed for insertion of y_2 in the chain of 3 items x_1, x_2 and y_1 already ranked. Finally we need an average of $H(7)$ for inserting y_3 in the chain of 6 items $x_1, x_2, x_4, x_6, y_1, y_2$ already ranked. In a similar manner for $n = 9$ we find that

$$(3.55) \qquad \begin{cases} e_9(1) = e_8(1) = H(3),\ e_9(2) = e_8(2),\ e_9(3) = H(8), \text{ and} \\ e_9(4) = \tfrac{1}{9} H(7) + \tfrac{8}{9} H(8) \ . \end{cases}$$

Then for $n \geq 1$

$$(1.4) \qquad T(n) = [\tfrac{n}{2}] + T([\tfrac{n}{2}]) + \sum_{c=1}^{[\frac{n-1}{2}]} s_n(c)$$

and $T(1) = 0$.

The values of $s_n(r)$ for small n are easy to compute but for large n (say $n > 10$) and large r (say $r > 7$) the calculations are tedious. For example, we have for $n = 5$

$$s_5(1) = H(4), \; s_5(2) = \tfrac{1}{2} H(3) + \tfrac{1}{2} H(4) \; \text{and} \; s_5(3) = H(4)$$

where $H(n)$ is given by (1.3). To see this we consider the configuration

It is clear that $s_5(1) = H(4)$ which is the expected number of comparisons needed for insertion of y_1 into the chain [illegible] under the P_5 procedure. If $x_1 < y_1$ (an event with probability $\frac{1}{2}$) then $H(4)$ is the expected number of comparisons for insertion of y_2 in the chain x_1 [illegible]. If $y_1 < x_1$ (an event with probability $\frac{1}{2}$) then $H(3)$ comparisons is the average needed for insertion of y_2 in the chain of 4 items x_2, x_3 and y_1 already ranked. Finally we need an average of $H(4)$ for insertion of y_3 in the chain of 6 items x_1, x_2, x_3, y_1, y_2 already ranked. In a similar manner for $n = 6$ we find that

$$(1.5) \qquad s_6(1) = s_6(1) = H(5), \; s_6(2) = s_6(2), \; s_6(3) = H(5), \; \text{and}$$

$$s_6(4) = \tfrac{2}{6} H(4) + \tfrac{4}{6} H(5) .$$

Define an integral valued function $j = j(n)$ for $n \geq 2$ by

$$(3.56) \qquad 2t_j \leq n < 2t_{j+1}$$

where t_j is given by (3.3). It is not difficult to see that for n even and all $\alpha \geq 1$

$$(3.57) \qquad e_n(\alpha) = e_{n-1}(\alpha) \ ;$$

and for $1 \leq \alpha \leq t_j-1$ and all n

$$(3.58) \qquad e_n(\alpha) = e_{n+1}(\alpha).$$

Using (3.57) and (3.58) it follows from (3.54) that for $k \geq 1$

$$(3.59) \qquad \begin{cases} F(2k+1) - F(2k) = \sum_{\alpha=t_j}^{k} e_{2k+1}(\alpha) - \sum_{\alpha=t_j}^{k-1} e_{2k}(\alpha) \\ \\ F(2k) - F(2k-1) = 1 + F(k) - F(k-1) \end{cases}$$

where for convenience we set for $t_j > k - 1$

$$(3.59a) \qquad \sum_{\alpha=t_j}^{k-1} e_{2k}(\alpha) = 0.$$

Here the boundary conditions for F are

$$(3.60) \qquad F(1) = F(0) = 0.$$

The values of $F(n)$ for $2 \leq n \leq 9$ are given in Table 2, Section (3.4). The following values of $e_n(\alpha)$ are needed to extend Table 2 for $10 \leq n \leq 16$. Since $2t_3 = 10$ and $2t_4 = 22$, then for all $n = 10,11,\ldots, 16$ we have $j = 3$ and $t_j = 5$.

Define an integral-valued function $j = j(n)$ for $n \geq 8$ by

$$(3.56) \qquad 2t_j \leq n < 2\,2t_{j+1}$$

where t_j is given by (3.?). It is not difficult to see that for n even and all $\nu \geq 1$

$$(3.57) \qquad e_n(\nu) = e_{n-1}(\nu) \; ;$$

and for $1 \leq \nu \leq t_j - 1$ and all n

$$(3.58) \qquad e_n(\nu) = e_{n+1}(\nu).$$

Using (3.57) and (3.58) it follows from (3.4?) that for $k \geq 1$

$$F(2k+1) - F(2k) = \sum_{\nu=t_j}^{k} e_{2k+1}(\nu) - \sum_{\nu=t_j}^{k-1} e_{2k}(\nu)$$

(3.59)

$$F(2k) - F(2k-1) = 1 + F(k) - F(k-1)$$

Here for convenience we set for $t_j > k - 1$

$$(3.59a) \qquad \sum_{\nu=t_j}^{k-1} e_{2k}(\nu) = 0.$$

Here the boundary conditions for F are

$$(3.60) \qquad F(1) = F(0) = 0.$$

The values of $F(n)$ for $2 \leq n \leq 9$ are given in Table 2, Section (3.4). The following values of $e_\nu(n)$ are needed to extend Table 2 for $10 \leq n \leq 16$. Since $2t_3 = 10$ and $2t_4 = 22$, then for all $n = 10, 11, \ldots, 16$ we have $j = 3$ and $t_j = 5$.

$$
(3.61)\quad \begin{cases} e_{11}(5) = H(11) = e_{12}(5) \\ e_{13}(5) = H(12) = e_{14}(5),\ e_{13}(6) = \frac{1}{13} H(11) + \frac{12}{13} H(12) = e_{14}(6) \\ e_{15}(5) = H(13),\ e_{15}(6) = \frac{1}{15} H(12) + \frac{14}{15} H(13) \\ e_{15}(7) = \frac{3}{13 \cdot 15} H(11) + \left(\frac{12 \cdot 2}{13 \cdot 15} + \frac{12}{13 \cdot 15}\right) H(12) + \frac{12 \cdot 13}{13 \cdot 15} H(13). \end{cases}
$$

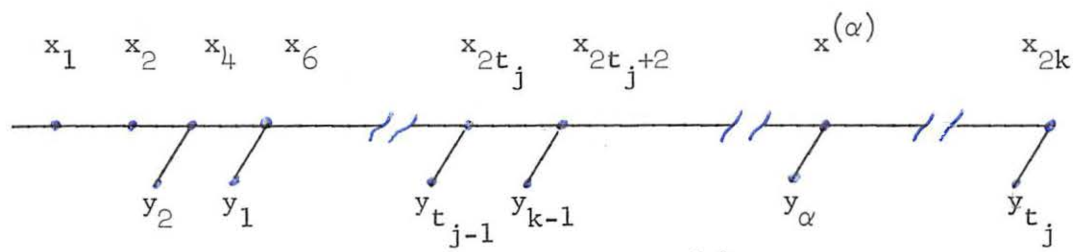
Table 3

Expected number of comparisons under the R_{FJ} procedure

(continuation of Table 2 above)

n	10	11	12	13	14	15	16
F(n)	$21\frac{268}{315}$	$25\frac{1373}{3465}$	$29\frac{218}{3465}$	$32\frac{37904}{45045}$	$36\frac{31469}{45045}$	$40\frac{29418}{45045}$	$44\frac{29418}{45045}$

Now we want to obtain more specific results about $e_n(\alpha)$. For $n = 2k$ (with $2t_j < n < 2t_{j+1}$) the diagram, after the 1st and 2nd step of R_{FJ} procedure, is:



For every α such that $t_j < \alpha \leq k-1$ let $x^{(\alpha)}$ be the item directly larger than y_α and let $N(x^{(\alpha)})$ denote the number of items from the set $\{y_{t_j}, y_{t_j+1}, \ldots, y_{\alpha-1}\}$ that are smaller than $x^{(\alpha)}$. Define

$$
(3.62)\qquad p_\lambda(\alpha, 2k) = \text{Prob}\ \{N(x^{(\alpha)}) = \lambda | R_{FJ}\}
$$

$\lambda = 0, 1, 2, \ldots, \alpha - t_j$. For convenience let $p_0(t_j, 2k) = 1$. Then $\sum_{\lambda=0}^{\alpha-t_j} p_\lambda(\alpha, 2k) = 1$ for all α with $t_j \leq \alpha \leq k-1$. Hence

$$
(3.63)\qquad e_{2k}(\alpha) = \sum_{\lambda=0}^{\alpha-t_j} p_\lambda(\alpha, 2k)\, H(2t_j + k - \alpha + \lambda).
$$

$$e_{11}(5) = E(11) = e_{12}(5)$$

$$e_{13}(6) = E(12) = e_{14}(6), \quad e_{13}(7) = \frac{1}{13}E(11) + \frac{12}{13}E(12) = e_{14}(7) \qquad ([illegible].01)$$

$$e_{15}(8) = E(13), \quad e_{15}(9) = \frac{1}{13}E(12) + \frac{12}{13}E(13)$$

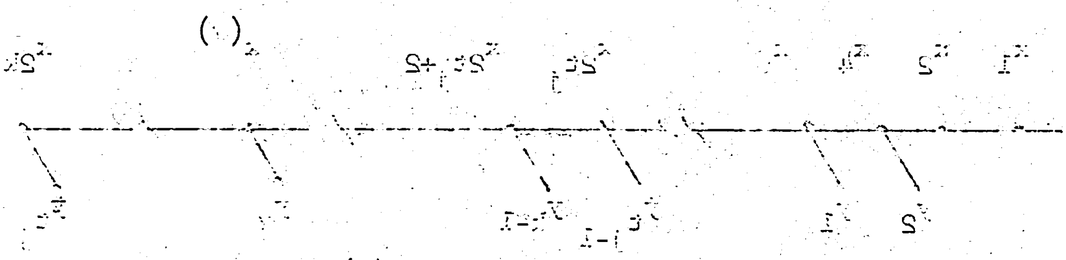
$$e_{15}(7) = \frac{[illegible]}{13\cdot 15}E(11) + \left(\frac{12\cdot 2}{13\cdot 14} + \frac{12}{13\cdot 14}\right)E(12) + \frac{12\cdot 11}{13\cdot 14}E(13).$$

Table [illegible]

Expected number of comparisons under the R_{E1} procedure
(continuation of Table 2 above)

n	10	11	12	13	14	15	16
$T(n)$	21 $\frac{203}{[illegible]}$	2[illegible] $\frac{1[illegible]}{[illegible]65}$	28 $\frac{218}{[illegible]}$	3[illegible] $\frac{904}{[illegible]}$	[illegible] $\frac{[illegible]}{[illegible]}$	40 $\frac{2[illegible]}{[illegible]}$	44 $\frac{2[illegible]}{4[illegible]}$

Now we want to obtain more specific results about $e_n(x)$. For $n = 2k$ (with $2t_j < x < 2t_{j+1}$) the diagram, after the 1st and 2nd step of R_{E1} procedure, is:



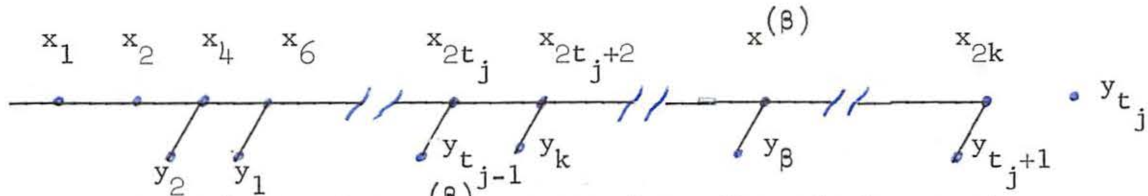
For every ν such that $t_j \le \nu \le k-1$ let $x^{(\nu)}$ be the item directly larger than y_ν and let $H(x^{(\nu)})$ denote the number of items from the set $y_{t_j}, y_{t_j+1}, \ldots, y_{k-1}$ that are smaller than $x^{(\nu)}$. Define

$$p_\nu(x, 2k) = \text{Prob}\,(H(x^{(\nu)}) = \nu - t_j) \qquad (1.02)$$

$\nu = 0, 1, 2, \ldots, k - t_j$. For convenience let $p_0(t_j, 2k) = 1$. Then $\sum_{\nu=0}^{k-t_j} p_\nu(x, 2k) = 1$ for all x with $t_j \le \nu \le k-1$. Hence

$$e_{2k}(x) = \sum_{\nu=0}^{k-t_j} p_\nu(x, 2k)\, E(2t_j + k - \nu + 1). \qquad ([illegible])$$

Similarly for $n = 2k + 1$ (with $2t_j < n < 2t_{j+1}$) the diagram is



For every $t_j < \beta \leq k$ let $x^{(\beta)}$ be the item directly larger than y_β and let $N(x^{(\beta)})$ denote the number of items from the set $\{y_{t_j}, y_{t_j+1}, \ldots, y_{\beta-1}\}$ that are smaller than $x^{(\beta)}$. Define

$$(3.64) \qquad p_\nu(\beta, 2k+1) = \mathrm{Prob}\{N(x^{(\beta)}) = \nu | R_{FJ}\}$$

$\nu = 0, 1, \ldots, \beta - t_j$. For convenience let $p_0(t_j, 2k+1) = 1$. Then $\sum_{\nu=0}^{\beta-t_j} p_\nu(\beta, 2k+1) = 1$ for all β with $t_j \leq \beta \leq k$. Hence

$$(3.65) \qquad e_{2k+1}(\beta) = \sum_{\nu=0}^{\beta-t_j} p_\nu(\beta, 2k+1)\, H(2t_j + k - \beta + \nu + 1).$$

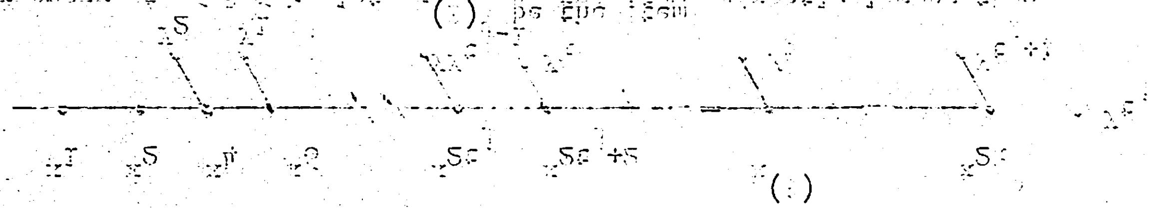
In (3.61) few values of $e_{2k}(\alpha)$ and $e_{2k+1}(\beta)$ are given. Substituting (3.63) and (3.65) in (3.59) and letting $\alpha - \lambda = i$ and $\beta - \nu = m$ we obtain

$$(3.66) \qquad \begin{cases} F(2k+1) - F(2k) = \displaystyle\sum_{m=t_j}^{k} \Big(\sum_{\beta=m}^{k} p_{\beta-m}(\beta, 2k+1)\Big) H(2t_j + k + 1 - m) \\ \qquad\qquad - \displaystyle\sum_{i=t_j}^{k-1} \Big(\sum_{\alpha=i}^{k-1} p_{\alpha-i}(\alpha, 2k)\Big) H(2t_j + k - i) \\ F(2k) - F(2k-1) = 1 + F(k) - F(k-1). \end{cases}$$

Since $F(2k+1) - F(2k)$ is not a simple expression it is useful to obtain bounds for it. Since $H(x)$ is an increasing function of x it follows from (3.63) and (3.65) that

$$(3.67) \qquad H(2t_j + h - \alpha) \leq e_{2k}(\alpha) \leq H(t_j + k)$$

Similarly for $n = 2k + 1$ (with $2t_j < n < 2t_{j+1}$) the diagram is



For every $t_j < \beta \le k$ let $x_{(\beta)}$ be the item directly larger than y_β and let $N(x_{(\beta)})$ denote the number of items from the set $\{y_{t_j}, y_{t_j+1}, \ldots, y_{\beta-1}\}$ that are smaller than $x_{(\beta)}$. Define

(4.94) $p_\nu(\beta, 2k+1) = \text{Prob}\{N(x_{(\beta)}) = \nu\}$,

$\nu = 0, 1, \ldots, \beta - t_j$. For convenience let $p_0(t_j, 2k+1) = 1$. Then $\sum_{\nu=0}^{\beta-t_j} p_\nu(\beta, 2k+1) = 1$ for all β with $t_j \le \beta \le k$. Hence

(4.95) $$e_{2k+1}(n) = \sum_{\nu=0}^{\beta-t_j} p_\nu(\beta, 2k+1)\, H(2t_j + [illegible] + \nu + 1).$$

In (4.91) for values of $e_{2k}(n)$ and $e_{2k+1}(n)$ are given. Substituting (4.94) and (4.95) in (4.7) and letting [illegible] we obtain

$$F(2k+1) - F(2k) = \sum_{\beta=t_j}^{k} \sum (\ldots p (\ldots, 2k+1)) H(2t_j + k + 1 - m)$$

(4.96) $$- \sum_{\beta=t_j}^{k-1} \sum (\ldots p (\ldots, 2k)) H(2t_j + k - \lambda)$$

$$F(2k) - F(2k-1) = 1 + F(k) - F(k-1).$$

Since $F(2k+1) - F(2k)$ is not a simple expression it is used to obtain bounds for it. Since $H(x)$ is an increasing function of x it follows from (4.94) and (4.95) that

(4.97) $H(2t_j + k - \lambda) \le e_{2k}(n) \le H(t_j + n)$

and

$$(3.68) \qquad H(2t_j + k - \beta + 1) \leq e_{2k+1}(\beta) \leq H(t_j + k + 1).$$

Using (3.67) and (3.68) we obtain from (3.59)

$$(3.69) \qquad \sum_{\beta=t_j}^{k} H(2t_j + k - \beta + 1) - (k - t_j)H(t_j + k) \leq F(2k + 1) - F(2k)$$

$$\leq (k + 1 - t_j)H(t_j + k + 1) - \sum_{\alpha=t_j}^{k-1} H(2t_j + k - \alpha).$$

It is interesting to note that from the definition of R_{FJ} procedure

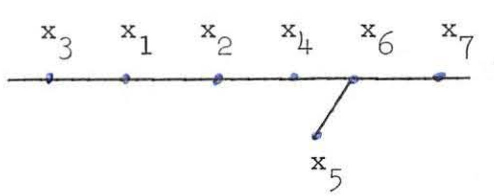
$$(3.70) \qquad F(n + 1) - F(n) = H(n + 1) \quad \text{provided} \quad n = 2t_j.$$

The relation (3.70) is also implied by (3.69); because for $n = 2t_j$, i.e., $k = t_j$, both lower and upper sides of (3.69) are equal to $H(2t_j + 1)$ and hence (3.70) follows. Hence the bounds in (3.69) can not be improved.

APPENDIX

Remarks on Trees Considered in the Text

For all of our trees (in particular, for the tree in Fig. 2) the symbol $H(m)$ at the end of a branch at any level means that there is a simple continuation of that branch which requires $H(m)$ additional comparisons on the average, where $H(m)$ is given by (1.5). In this simple continuation we have to rank 1 item among $m - 1$ items already ranked, and the procedure used for this purpose is R_S described in Section (2.1). In Chapter 2 is proved that R_S is an E-optimal (M-optimal) continuation and it requires $H(m)$ comparisons in average (maximum of $\{H(m)\}$ comparisons). For example, the continuation $H(5)$ for the branch marked (*) in Fig. 2 is given in Fig. (A.1) below. For this $H(5)$-situation we have the diagram



which indicates that

$x_3 < x_1 < x_2 < x_4 < x_6 < x_7$ and $x_5 < x_6$; and the continuation:

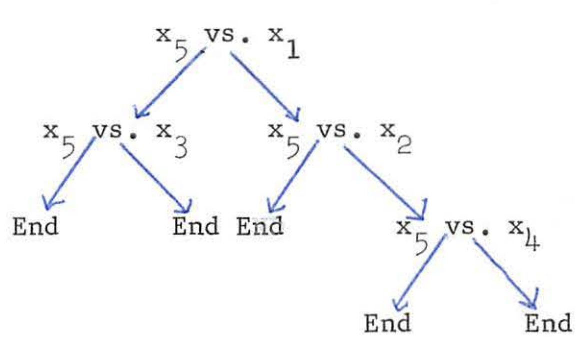
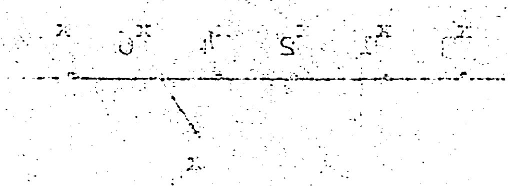


Fig. (A.1). <u>Continuation of the branch marked (*) in Fig. 2.</u>

APPENDIX

Remarks on Trees Considered in the Text

For all of our trees (in particular, for the tree in Fig. 2) the symbol H(m) at the end of a branch at any level means that there is a simple continuation of that branch which requires H(m) additional comparisons on the average, where H(m) is given by (1.5). In this simple continuation we have to rank 1 item among m - 1 items already ranked, and the procedure used for this purpose is R_1 described in section (2.1). In Chapter 2 it is proved that R_1 is an M-optimal (N-optimal) continuation and it requires H(m) comparisons in average (minimum of H(m) comparisons). For example, the continuation H(5) for the branch marked (*) in Fig. 2 is given in Fig. (A.1) below. For this H(5)-situation we have the diagram



which indicates that

$x_3 < x_1 < x_2 < x_4 < x_0 < x_6$ and $x_5 > x_1$; and the continuation:

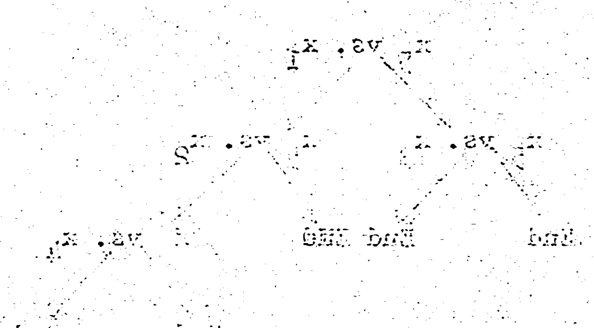
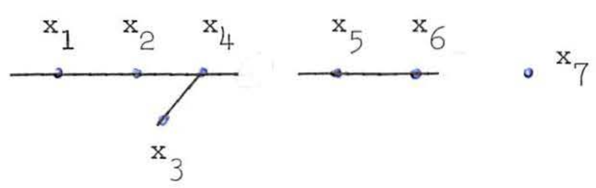


Fig. (A.1). Continuation of the branch marked (*) in Fig. 2.

The parenthetical '(315 cases)' at the root of the tree in Fig. 2 is the number of permutations of 7 items $x_i (i = 1,2,\ldots, 7)$ consistent with the conditions

$$x_1 < x_2 < x_4,\ x_3 < x_4,\ x_5 < x_6,$$

which can also be represented by the diagram A.2.



A.2

The diagram A.2 indicates the results of complete pairing of 7 items. Other numbers at different stages have a similar meaning, i.e., they represent the number of permutations of 7 items subject to the inequalities known at that stage.

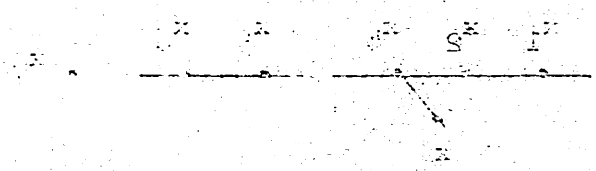
The word 'same' written at the end of some branches means that the continuation of that branch is essentially (except for the renaming of some items) the same as the continuation of the branch on the same level and the same structure (diagram), and consequently with the same number of cases.

The numbers in a circle, which appear only when they are non-zero, indicates the number of powers of 2 between two numbers accompanying the arrows. For example, the encircled number 1 between the arrows accompanying the partition (84, 63) indicates that there is exactly one power of 2 (namely $2^6 = 64$) between 84 and 63. The partition (84, 63) is the result of the x_3 vs. x_2 comparison which divides the 147 states of nature into 2 sets depending on whether $x_3 < x_2$ or $x_3 > x_2$.

The permutational '(?? cases)' at the root of the tree in Fig. 2 is the number of permutations of 7 items x_i (i = 1,2,...,7) consistent with the conditions

$$x_1 < x_2 < x_4,\ x_3 < x_5,\ x_6 < x_7$$

which can also be represented by the diagram A.2.



A.2

The diagram A.2 indicates the results of complete-ordering of 7 items. Other numbers at different stages have a similar meaning, i.e., they represent the number of permutations of 7 items subject to the inequalities known at that stage.

The word 'same' written at the end of some branches means that the continuation of that branch is essentially (except for the renaming of some items) the same as the continuation of the branch on the same level with the same structure (diagram), and consequently with the same number of cases.

The numbers in a circle, which appear only when they are non-zero, indicate the number of powers of 2 between two numbers accompanying the arrows. For example, the encircled number 1 between the arrows accompanying the partition (34, 63) indicates that there is exactly one power of 2 (namely $2^6 = 64$) between 34 and 63. The partition (34, 63) is the result of the x_1 vs. x_2 comparison which divides the 1?? states of nature into 2 sets depending on whether $x_1 < x_2$ or $x_1 > x_2$.

To compute the expected length of the tree in Fig. 2 we use a simple formula due to Sobel (see (6.13) in [16]), putting in 3 for the number of noise units and using (1.5). This formula gives for any tree T the expected length $E(L|T)$

$$E(L|T) = H(m) + \frac{U}{m} = 315 + \frac{3}{315} = 8 + \frac{121}{315} \tag{A.3}$$

where m is the number of cases at the root of the tree. Finally, since we made 4 comparisons prior to the tree in Fig. 2, the expected number of comparisons for ranking 7 items according to the procedure R_1 is

$$A(7|R_1) = 12 + \frac{121}{315} . \tag{A.4}$$

The introduction of $H(m)$-situation greatly simplifies a tree. For example, with this notation the tree in Fig. 1 takes the simple form

$$\begin{tikzcd} & x_1 \text{ vs. } x_3 \text{ (6 cases)} \arrow[dl] \arrow[dr] & \\ H(3) & & H(3) \end{tikzcd}$$

and by (1.5) and (A.3) with $U = 0$ and $m = 6$ the expected length is $H(6) = 2 + \frac{2}{3}$.

To compute the expected length of the tree in Fig. 2 we use a simple formula due to Jonsi (see (6.1) in [10]), putting in [illegible] for the number of nodes units and using (1.5). This formula gives for any tree T the expected length $E(L;T)$

$$(A.3) \qquad E(L|T) = N(a) + \frac{U}{[illegible]} = 1[illegible] + \frac{[illegible]}{[illegible]} = [illegible] + \frac{121}{1[illegible]}$$

where [illegible] is the number of edges at the root of the tree. Finally, since we made 4 comparisons prior to the tree in Fig. 2, the expected number of comparisons for ranking [illegible] items according to the procedure R_1 is

$$(A.4) \qquad A([illegible]|R_1) = 12 + \frac{121}{15} .$$

The introduction of $U(\cdot)$-situation greatly simplifies a tree. For example, with this notation the tree in Fig. 1 takes the simple form:

n_1 vs. $n_{[illegible]}$ ([illegible] cases)

$E([illegible])$ $\qquad$ $E([illegible])$

and by (1.5) and (A.[illegible]) with $U = 0$ and [illegible] = [illegible] the expected total length is $E(G) = 2 + \frac{2}{3}$.

REFERENCES

[1] Busaker, R. G. and Saaty, T. L. (1965). The Graphs and Networks. McGraw-Hill, New York.

[2] Cesari, Y. (1967). Questionnaire, codage et tris. (These, Doctorat 3e Cycle) Faculte des Sciences de Paris, 35-39.

[3] Cramer, H. (1947). Mathematical Methods of Statistics. Princeton University Press, Princeton.

[4] David, H. A. (1963). The Method of Paired Comparisons. Charles Griffin and Co. Ltd., London.

[5] Ford Jr., L. R. and Johnson, S. M. (1959). A tournament problem, American Mathematical Monthly 66, 387-389.

[6] Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. Proceedings I. R. E. 9, 1098-1101.

[7] Iverson, K. E. (1962). A Programming Language. John Wiley, New York.

[8] Kislicyn, S. S. (1962). On a bound for the smallest average number of pairwise comparisons necessary for the complete ordering of n objects with different weights (Russian). Vestnik Leningrad Univ. (Series on Math., Mech., and Astron.) 18 No. 1, 162-163.

[9] Kislicyn, S. S. (1963). A sharpening of the bound on the smallest average number of comparisons necessary for the complete ordering of a finite set (Russian). Vestnick Leningrad Univ. (Series on Math., Mech. and Astron.) 19 No. 4, 143-145.

[9a] Moon, J. W. (1968). Topics on Tournaments. Holt, Rinehart and Winston New York.

[10] Ogilvy, C. S. (1962). Tomorrow's Math. Oxford University Press, New York.

[11] Picard, C. (1965). Theorie des Questionnaires, Gauthiers-Villars, Paris.

REFERENCES

[1] Busacker, R. G. and Saaty, T. L. (1965). Finite Graphs and Networks. McGraw-Hill, New York.

[2] Cesari, Y. (1967). Questionnaire, codage et tris. (Thesis, Doctorat 3e Cycle) Faculté des Sciences de Paris. 47-49.

[3] Cramér, H. (1947). Mathematical Methods of Statistics. Princeton University Press, Princeton.

[4] David, H. A. (1963). The Method of Paired Comparisons. Charles Griffin and Co. Ltd., London.

[5] Ford Jr., L. R. and Johnson, S. M. (1959). A tournament problem. American Mathematical Monthly 66, 387-389.

[6] Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. Proceedings I. R. E. 40, 1098-1101.

[7] Iverson, K. E. (1962). A Programming Language. John Wiley, New York.

[8] Kislicyn, S. S. (1962). On a bound for the smallest average number of pair-wise comparisons necessary for the complete ordering of n objects with different weights (Russian). Vestnik Leningrad Univ. (Series on Math., Mech. and Astron.) 17 No. 1, 162-163.

[9] Kislicyn, S. S. (1963). A refinement of the bound on the smallest average number of comparisons necessary for the complete ordering of a finite set (Russian). Vestnik Leningrad Univ. (Series on Math., Mech. and Astron.) 18 No. 4, 14-15.

[10] Ogilvy, C. S. (1962). Tomorrow's Math. Oxford University Press, New York.

[11] Picard, C. (1965). Théorie des Questionnaires. Gauthier-Villars, Paris. New York.

[12] Picard, C. (1972). Graphes et Questionnaires. Gauthier-Villars, Paris.

[12] Sandelius, M. (1961). On an optimal search problem, Amer. Math. Monthly 68, 134-138.

[13] Schreier, J. (1932). On tournament elimination systems (Polish), Mathesis Polska, 154-160. (Radcliffe Library, Oxford).

[14] Sobel, Milton (1968). Binomial and hypergeometric group-testing. Studia Scientiarum Mathematicarum Hungarica III, 19-42.

[15] Sobel, Milton (1966). Optimal group-testing. University of Minnesota, Technical Report No. 83.

[16] Sobel, Milton (1968). On an optimal search problem for t best using binary errorless comparisons. University of Minnesota, Technical Report No. 113.

[17] Steinhaus, H. (1950 and 1960). Mathematical Snapshots, Oxford Univ. Press, New York (See pp. 37-40 in 1950 edition and 53-55 in 1960 edition.)

[18] Steinhaus, H. (1963). One Hundred Problems in Elementary Mathematics, 26-117. Pergamon Press, Oxford. (English translation of 1958 Polish edition.)

[19] Steinhaus, H. (1959). Some remarks about tournaments, Calcutta Math. Soc. Golden Jubilee Comm. Vol (1958/1959), Part II, 323-327 (MR27 No. 4770).

[12] Sandelius, M. (1961). On an optimal search problem. Amer. Math. Monthly 68, 133-134.

[13] Schreier, J. (1932). On tournament elimination systems (Polish). Mathesis Polska, 7, 154-160. (Radcliffe Library, Oxford).

[14] Sobel, Milton (1960). Binomial and hypergeometric group-testing. Studia Scientiarum Mathematicarum Hungarica III, 19-42.

[15] Sobel, Milton (1967). Optimal group-testing. University of Minnesota. Technical Report No. 95.

[16] Sobel, Milton (1967). On an optimal search problem for a best using binary errorless comparisons. University of Minnesota, Technical Report No. 113.

[17] Steinhaus, H. (1950 and 1960). Mathematical Snapshots, Oxford Univ. Press, New York (see pp. 37-40 in 1950 edition and 55-56 in 1960 edition).

[18] Steinhaus, H. (1964). One Hundred Problems in Elementary Mathematics. 29-117. Pergamon Press, Oxford. (English translation of 1958 Polish edition.)

[19] Steinhaus, H. (1958). Some remarks about tournaments, Calcutta Math. Soc. Golden Jubilee Comm. Vol (1958/1959), Part II. 323-327 (MR 22 No. 4670).